FLASH® LITE™ 2.x und 3.0 -Anwendungen entwickeln



Inhalt

Kapitel 10: Einführung in Flash Lite 2.x und 3.x ActionScript

Kapitel 11: Warnungen und Fehlermeldungen

Kapitel 1: Flash Lite 2.x und 3.x - Überblick

Adobe® Flash® Lite™ ist eine speziell für Mobilgeräte entwickelte Version von Flash® Player. In dieser Dokumentation werden Macromedia® Flash® Lite™ 2.0 und Macromedia Flash Lite 2.1 von Adobe behandelt, die gemeinsam als Flash Lite 2.x bezeichnet werden. Adobe Flash Lite 3.0 und 3.1, gemeinsam als Flash Lite 3.x bezeichnet, werden ebenfalls behandelt.

Flash Lite – Grundlagen

Informationen zu Flash Lite

Macromedia® Flash Lite 1.0 und Macromedia Flash Lite 1.1 basieren auf dem Macromedia Flash Player 4. Flash Lite 2.0 und 2.1 (gemeinsam als 2.x bezeichnet) basieren auf dem Macromedia Flash Player 7 und unterstützen die meisten (jedoch nicht alle) Funktionen von Flash Player 7. Flash Lite 2.x enthält außerdem spezielle Funktionen für Mobilgeräte, die in Flash Player 7 nicht verfügbar sind. Beispielsweise können Sie in Flash Lite 2.x gerätespezifische Medientypen (Bilder, Sounds, Videos) laden, die nicht nativ von Flash Lite unterstützt werden. Flash Lite 2.x bietet außerdem Funktionen zur Geräteintegration, wie etwa Möglichkeiten für Telefonanrufe und Kurzmeldungen.

Flash Lite 3.0 basiert auf Flash Player 7, und mit dieser Version werden die Unterstützung für Flash Video (FLV) und für das Durchsuchen der meisten Flash 8-Inhalte sowie zahlreiche Leistungsverbesserungen eingeführt. Flash Lite 3.0 enthält außerdem ein neues Sicherheitsmodell, das dem in Flash Player 8 verwendeten entspricht.

Flash Lite 3.1 basiert ebenfalls auf Flash Player 7. Diese Version bietet Funktionen, mit denen die Benutzer von mobilen Geräten leichter im Web navigieren können. Viele dieser Funktionen waren bereits in Flash® Player 8 vorhanden, wurden aber in Flash Lite nicht unterstützt. Jetzt werden diese Funktionen in Flash Lite 3.1 unterstützt. Flash Lite 3.1 ist in der Lage, ActionScript™ 2.0-Inhalte von Flash 9-Websites wiederzugeben. Die Verarbeitung und Wiedergaben des mit Flash Player 9 eingeführten ActionScript™ 3.0 werden von Flash Lite 3.1 aber nicht unterstützt. Weitere Informationen finden Sie unter „Flash Lite 3.1-Funktionen“ auf Seite 2.

Komponenten in Flash Lite

Die mit Flash installierten Komponenten (z. B. „DataGrid“ und „Accordion“) sind für die Verwendung in Flash-Desktop-Anwendungen ausgelegt. Aufgrund der Speicher- und Leistungsanforderungen ist die Verwendung in Flash Lite-Anwendungen normalerweise nicht möglich. Adobe empfiehlt, die Benutzeroberflächen-Standardkomponenten in Flash Lite-Anwendungen nicht zu verwenden.

Optimieren von Inhalten auf Leistung und Speichernutzung

Beim Optimieren von Flash-Inhalten müssen Sie einige Grundsätze beachten. Flash-Entwickler mussten beispielsweise sehr komplexe Grafiken, übermäßiges Tweening und die übermäßige Verwendung von Transparenz vermeiden.

In früheren Version von Flash wurden die meisten dieser Leistungsprobleme für den Desktop behoben, Adobe Flash Lite-Entwickler sehen sich jedoch noch immer mit zusätzlichen Herausforderungen aufgrund der Beschränkungen von Mobilgeräten konfrontiert: Die Leistungsunterschiede zwischen Geräten können erheblich sein und da das Authoring für Mobilgeräte häufig ein Veröffentlichen für viele verschiedene Geräte erfordert, müssen sich Entwickler manchmal an den kleinsten gemeinsamen Nenner halten.

Bei der Optimierung von Inhalten für mobile Geräte sind häufig Kompromisse erforderlich. Beispielsweise könnte eine Technik besser aussehen, während eine andere eine bessere Leistung zeigt. Beim Abwägen der Kompromisse werden Sie wiederholt zwischen dem Testen im Emulator und dem Testen auf dem Zielgerät wechseln müssen.

Bei Flash 8 war der Emulator für Flash Lite 2.x ein Teil der Flash-Authoring-Umgebung. Ab Flash CS3 und Flash Lite 3.0 sowie Flash CS4 und Flash Lite 3.1 ist die Emulator-Funktion Teil von Adobe® Device Central. Device Central gestattet es, Flash Lite-Projekte auf einer Vielzahl von Geräten zu emulieren; dabei kann die Geräteanzeige, die Speichernutzung und die Leistung auf bestimmten Geräten emuliert werden. Umfassende Informationen über den Einsatz des Emulators bei der Optimierung Ihrer Flash Lite-Inhalte für mobile Geräte finden Sie in der Dokumentation zu Device Central im Abschnitt über empfohlene Verfahren bei der Erstellung von Inhalten für mobile Geräte.

Flash Lite 3.1-Funktionen

Verbessertes Web-Browsing

Flash Lite 3.1 bietet eine Reihe von Funktionen, mit denen die Benutzer von mobilen Geräten leichter im Web navigieren können. Viele dieser Funktionen waren bereits in Flash® Player 8 vorhanden, wurden aber in Flash Lite nicht unterstützt. Jetzt werden diese Funktionen in Flash Lite 3.1 unterstützt. Diese neuen Funktionen werden nachstehend beschrieben.

Flash 9-Browsing

Flash Lite 3.1 wurde mit Unterstützung für Flash Player 8 veröffentlicht und unterstützt somit das mit Flash Player 9 eingeführte ActionScript™ 3.0 nicht. Viele Flash 9-Websites enthalten aber ActionScript 2.0-Inhalte, die in Flash Lite 3.0 nicht wiedergegeben werden können. Flash Lite 3.1 kann diese Inhalte von Flash 9-Websites wiedergeben. Der Flash Lite 3.1-Player kann Flash 9-Inhalte verarbeiten. Beim Laden von Websites mit Flash 9-Inhalten werden von Flash Lite 3.1 die SWF-Datei ausgewertet und nur die Flash 9-Inhalte wiedergegeben, die ActionScript 2.0 verwenden. Verwenden die Flash 9-Inhalte ActionScript 3, werden diese von Flash Lite 3.1 nicht wiedergeben. Es erscheint ein benutzerfreundliches Fehlersymbol (wie bei Flash Lite 3.0).

LocalConnection-Objekt

Mit der LocalConnection-Klasse können Sie eine oder mehrere SWF-Dateien entwickeln, die ohne `fscommand`() oder JavaScript untereinander Anweisungen versenden können. LocalConnection-Objekte können nur mit SWF-Dateien, die auf demselben Client-Gerät ausgeführt werden, kommunizieren. Sie können jedoch in verschiedenen Anwendungen ausgeführt werden. LocalConnection-Objekte können auch zum Senden und Empfangen von Daten in einer einzelnen SWF-Datei verwendet werden. Diese Fähigkeit wird aber nicht oft verwendet. Die Flash Lite 3.1-Implementierung des LocalConnection-Objekts unterstützt die Kommunikation zwischen SWF-Dateien, die mit demselben Prozess (zum Beispiel im selben Browserfenster) ausgeführt werden.

HTML-Textverarbeitung

Flash Lite 3.1 unterstützt jetzt verschiedene HTML-Textbearbeitungstags, die in Flash Lite 3.0 nicht unterstützt wurden. Flash Lite 3.1 unterstützt die Verwendung von Text als Links, das `<img>` -Tag und die Verarbeitung von Cascading Style Sheets (CSS).

Text als Links Das jetzt in Flash Lite 3.1 unterstützte `<a>`-Tag dient zum Erstellen von Hyperlinks. Vollständige Informationen über die Attribute und Verwendung des `<a>`-Tags finden Sie in der Flash Lite-Dokumentation.

Eingebettete externe Bilder Textfelder in Flash Lite 3.1 unterstützen das `<img>`-Tag. Dieses Tag kann auf Bilder verweisen, die im lokalen Dateisystem oder im Netzwerk gespeichert sind. Das Bild kann das Format JPG, GIF oder

PNG aufweisen und wird wiedergegeben, wenn die Plattform diesen Bildtyp unterstützt. Das Textfeld kann verschiedene Attribute aufweisen wie beispielsweise „mehrzeilig“ oder „Zeilenumbruch“, die sich auf die Textwiedergabe auswirken. Das `<img>`-Tag kann nicht nur auf Bilder, sondern auch auf eine SWF-Datei oder ein MovieClip-Objekt verweisen. Sind diese Medien in das Textfeld eingebettet, gelten sie als untergeordnete Elemente des Textfelds und werden mit ActionScript gesteuert und manipuliert.

CSS-Unterstützung Die Textformatierung mit CSS-Stilen wird in Flash Lite 3.0 nicht unterstützt. In Flash Player auf dem Desktop angezeigte Flash 8-Inhalte unterstützen die Formatierung mit CSS-Stilen, die entweder als separate CSS-Stylesheet-Datei oder durch Hinzufügen von Inline-Stilen mittels ActionScript geladen werden. Die CSS-Funktion ermöglicht Entwicklern das Optimieren des Erscheinungsbilds und Verhaltens ihrer Anwendungen. Flash Lite 3.1 unterstützt diese Funktion.

Unterstützung von WMODE

Flash Lite 3.1 unterstützt jetzt den `wmode`-Parameter. Durch Verwendung dieses Parameters in Verbindung mit dem ActiveX-Plugin und einem Browser können Sie festlegen, ob die Hintergrundfarbe oder das Hintergrundbild der HTML-Seite, auf der der Flash-Film läuft, durchscheint, um Rich-Media-Inhalte mit DHTML-Inhalten zu kombinieren. `wmode` kann den Wert`window` (Standard), `opaque` oder`transparent` aufweisen:

- Bei Eingabe von `Window` wird die Anwendung in einem eigenen rechteckigen Fenster auf einer Webseite abgespielt. Diese Option legt fest, dass die -Anwendung keinerlei Interaktion mit den HTML-Ebenen durchführt und immer das oberste Element ist.
- Bei Eingabe von `Opaque` verdeckt die Anwendung alle Objekte, die sich hinter ihm auf der Seite befinden.
- Bei Eingabe von `transparent` scheint der Hintergrund der HTML-Seite an allen transparenten Stellen der Anwendung durch. Dabei kann sich jedoch die Animation verlangsamen.

Unterstützung von scale, salign und align

Flash Lite 3.1 unterstützt jetzt die JavaScript-Parameter `scale`, `salign` und `align`, die festlegen, wie auf einer HTML-Seite eingebettete Rich-Media-Inhalte skaliert, positioniert und ausgerichtet werden. Der Parameter `scale` definiert, wie die SWF-Datei im Browserfenster platziert wird, wenn die Breiten- und Höhenangaben Prozentzahlen sind. Die Werte können `showall`, `noborder`, oder `exactfit` sein. Das Attribut align bestimmt, wie die SWF-Datei im Browserfenster positioniert wird. Die Werte können `L` (links), `R` (rechts), `T` (oben) oder `B` (unten) sein. Der Parameter salign gibt an, wo in dem mit width und height definierten Bereich eine skalierte SWF-Datei positioniert wird. Die Werte können `TL` (oben links), `TR` (oben rechts), `BL` (unten links) oder `BR` (unten rechts) sein.

GetURL _target-Parameter

`_target` ist ein optionaler Parameter der globalen ActionScript-Funktion getURL, der das Fenster oder den HTML-Frame angibt, in das bzw. den das Dokument geladen werden soll. Der `_target`-Parameter wurde bisher nicht für Flash Lite-Anwendungen unterstützt, da die Browser mobiler Geräte bis vor kurzem mehrere Fenster nicht unterstützt haben. Sie können entweder eine leere Zeichenfolge eingeben oder einen der folgenden reservierten Zielnamen auswählen:

`_self` bezeichnet den aktuellen Frame im aktuellen Fenster.

`_blank` bezeichnet ein neues Fenster.

`_parent` bezeichnet den übergeordneten Frame des aktuellen Frames.

`_top` bezeichnet den obersten Frame im aktuellen Fenster. Wird `_target` nicht angegeben, lautet der Standardwert `_self`. Das Browserfenster muss Frames enthalten, wenn alle `_target`-Parameter wie angegeben funktionieren sollen. Enthält das Browserfenster keine Frames, funktioniert `_blank` wie erwartet, und die URL wird in einem neuen leeren Fenster geöffnet. Die URL wird allerdings auch mit dem `_self`-, `_parent`- und `_top`-Parameter im selben Fenster geöffnet.

Änderungen am Verhalten von SharedObject

Die Regeln für das Speichern von gemeinsam genutzten lokalen Objekten in Flash Lite 3.1 sind jetzt mit den Regeln für Flash Player 8 (Desktop) voll kompatibel. Vor Flash Lite 3.1 wurde der Speicherort eines gemeinsam genutzten Objekts vom Namen der SWF-Datei abgeleitet. B.swf konnte also nicht auf ein in A.swf gespeichertes Objekt zugreifen, wobei es keine Rolle spielte, wo sich die SWF-Dateien befanden. In Flash Lite 3.1 wird der Speicherort eines gemeinsam genutzten Objekts von seiner URL und nicht von seinem Namen abgeleitet. Somit kann /data/A.swf auf einem mobilen Gerät auf gemeinsam genutzte Objekte zugreifen, die in /B.swf gespeichert sind. Ein weiterer wesentlicher Unterschied zwischen Flash Lite 3.1 und früheren Versionen besteht darin, dass gemeinsam genutzte Objekte bei Verwendung einer früheren Version im Falle einer Änderung der SWF-Datei nicht mehr als dieselben Objekte angesehen wurden, selbst wenn der Name gleich blieb. In Flash Lite 3.1 wird das gemeinsam genutzte Objekt einer SWF-Datei so lange als dasselbe Objekt behandelt, solange die URL/Sandbox dieselbe ist, auch wenn die SWF-Datei geändert oder umbenannt wurde. Bei Versionen vor 3.1 konnten zwei verschiedene Versionen einer SWF-Datei auch nicht auf die gemeinsam genutzten Objekte der jeweils anderen Datei zugreifen. Bei Version 3.1 ist das möglich. Flash Lite 3.1 baut auf einigen wichtigen Verhaltensänderungen auf, die in Flash Lite 3.0 eingeführt wurden. Vor Flash Lite 3.0 konnten nur lokal gespeicherte SWF-Dateien auf gemeinsam genutzte Objekte von Flash Lite zugreifen. SWF-Dateien, die in einem netzwerkfähigen Browser wiedergegeben werden, können keine gemeinsamen Flash Lite-Objekte verwenden. Bei Flash Lite 3.0 und neueren Versionen können SWF-Dateien, die in der Netzwerksicherheits-Sandbox abgespielt werden, auf gemeinsam genutzte Objekte zugreifen und diese speichern. Ab Flash Lite 3.0 konnten auch alle Netzwerk-SWF-Dateien lokale gemeinsam genutzte Objekte speichern und auf diese zugreifen, wobei der Zugriff aber nur auf Pro-SWF-Basis möglich war. Bei Flash Lite 3.1 können alle SWF-Dateien aus derselben Netzwerkdomäne gegenseitig auf die gemeinsam genutzten Objekte zugreifen.

Optionaler Parameter von SharedObject.getLocal

Die SharedObject-Klasse umfasst eine `getLocal`-Methode mit zwei optionalen Parametern.

- `secure`: ein boolescher Parameter (mit Flash 8 eingeführt), den Sie zur Unterscheidung zwischen gemeinsam genutzten Objekten von HTTP- und HTTPS-Domänen verwenden können. Flash Lite 3.1 unterstützt jetzt den `secure`-Parameter.
- `localPath`: von Flash Lite nicht unterstützt.

Bild-API

Flash Lite 3.1 unterstützt APIs, die Anwendungsentwickler durch Versetzen, Drehen, Skalieren und Neigen von Bildern in mobilen Anwendungen zur Erhöhung der Ausdruckskraft verwenden. Die mit Flash 8 eingeführten Pakete flash.display und flash.geom bilden die Grundlage dieser neuen Funktion. Das Paket flash.display unterstützt die BitmapData-Klasse zur Erstellung von Bitmaps beliebiger Größe und flash.geom unterstützt Klassen und Methoden zur Bearbeitung auf Pixelebene einschließlich Versetzen, Drehen, Skalieren und Neigen. Im *Adobe Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch* wird die Verwendung dieser Pakete genau beschrieben.

flash.display.BitmapData-Klasse

Mit dieser Klasse können Sie beliebig große transparente oder deckende Bitmapbilder erstellen und während der Laufzeit auf verschiedene Weise manipulieren, wodurch die Bitmapwiedergabe von den internen Anzeigeaktualisierungsroutinen des Flash Lite Player getrennt wird. Sie können BitmapData-Objekte an MovieClip-Objekte anfügen, indem Sie die MovieClip.attachBitmap()-Methode verwenden. Die folgenden BitmapDataClass-Methoden werden von Flash Lite 3.1 teilweise oder nicht unterstützt. Alle anderen Methoden sind in der Flash Lite-Dokumentation beschrieben.

Teilweise unterstützte Methoden

Die folgenden BitmapDataClass-Methoden werden von Flash Lite 3.1 teilweise unterstützt.

Methode	Beschreibung
`copyPixels`	Bietet eine schnelle Routine zur Bearbeitung von Pixeln zwischen Bildern ohne Dehnung, Drehung oder Farbeffekte. Die Parameter `alphaBitmap`, `alphaPoint` oder `mergeAlpha` dieser Methode werden von Flash Lite 3.1 nicht unterstützt.
`draw`	Zeichnet ein Quellbild oder einen Quellmovieclip mithilfe des Flash Lite Player-Vektorrenderers in ein Zielbild. Der Parameter `blendmode` dieser Methode wird von Flash Lite 3.1 nicht unterstützt.

Nicht unterstützte Methoden

Die folgenden BitmapDataClass-Methoden werden von Flash Lite 3.1 nicht unterstützt.

Methode	Beschreibung
`applyFilter`	Erstellt anhand eines Quellbilds und eines filter-Objekts das gefilterte Bild. Da Flash Lite 3.1 keine Filter unterstützt, wird diese Methode nicht unterstützt.
`generateFilterRect`	Bestimmt das Zielrechteck, auf das sich der Aufruf der `applyFilter()`-Methode auswirkt. Die Angaben beziehen sich auf ein BitmapData-Objekt, ein Quellrechteck und ein filter-Objekt. Da Flash Lite 3.1 keine Filter unterstützt, wird diese Methode nicht unterstützt.
`noise`	Füllt ein Bild mit Pixeln, die zufällige Störungen darstellen. Diese Methode wird von Flash Lite 3.1 nicht unterstützt.
`paletteMap`	Ordnet die Farbkanalwerte in einem Bild mit maximal vier Arrays von Farbpalettendaten neu zu, wobei die einzelnen Arrays jeweils einen Kanal darstellen. Diese Methode wird von Flash Lite 3.1 nicht unterstützt.
`perlinNoise`	Erstellt ein Perlin-Störungsbild. Diese Methode wird von Flash Lite 3.1 nicht unterstützt.
`pixelDissolve`	Führt eine Pixelauflösung von einem Quellbild zu einem Zielbild oder anhand desselben Bilds durch. Diese Methode wird von Flash Lite 3.1 nicht unterstützt.
`scroll`	Führt für ein Bild einen Bildlauf anhand eines bestimmten Pixelbetrags (x, y) durch. Diese Methode wird von Flash Lite 3.1 nicht unterstützt.
`threshold`	Prüft die Pixelwerte eines Bilds für einen angegebenen Schwellenwert und weist den Pixeln, die die Bedingung des Tests erfüllen, neue Farbwerte zu. Diese Methode wird von Flash Lite 3.1 nicht unterstützt.

flash.geom.Point-Klasse

Diese Klasse gibt eine Position in einem zweidimensionalen Koordinatensystem an, in dem *x* die horizontale Achse und *y* die vertikale Achse darstellt. Diese Klasse wird von Flash Lite 3.1 voll unterstützt.

flash.geom.Matrix-Klasse

Diese Klasse stellt eine Matrix dar, die festlegt, wie Punkte eines Koordinatenraums einem anderen Koordinatenraum zugeordnet sind. Durch Einstellung der Eigenschaften eines Matrix-Objekts und deren Anwendung auf ein MovieClip- oder BitmapData-Objekt können Sie verschiedene grafische Transformationen an dem Objekt vornehmen. Diese Umformungsfunktionen umfassen Versetzen, Drehen, Skalieren und Neigen. Flash Lite 3.1 unterstützt sämtliche Methoden der Flash.geom.matrix-Klasse mit Ausnahme von createGradientBox.

flash.geom.ColorTransform-Klasse

Mit dieser Klasse können Sie mathematisch alle Farbwerte in einem Movieclip anpassen. Die auch als Farbtransformation bezeichnete Farbanpassungsfunktion kann auf alle vier Kanäle angewendet werden: Rot, Grün, Blau oder Alpha-Transparenz. Diese Klasse wird von Flash Lite 3.1 voll unterstützt.

flash.geom.Rectangle-Klasse

Diese Klasse wird vom BitmapData-Objekt zur Bearbeitung eines bestimmten Bereichs in einer Bitmap verwendet. Die Klasse ermöglicht die Bearbeitung eines oder mehrerer Rechtecke. Diese Klasse wird von Flash Lite 3.1 voll unterstützt.

flash.geom.Transform-Klasse

Diese Klasse erfasst Daten zu Farbtransformationen und Koordinatenänderungen für ein Movieclip-Objekt. Normalerweise erhalten Sie ein Transform-Objekt, indem Sie den Wert der transform-Eigenschaft von einem MovieClip-Objekt holen. Diese Klasse wird von Flash Lite 3.1 voll unterstützt.

setTimeout/clearTimeout

Die globalen ActionScript 2.0-Funktionen `setTimeout` und `clearTimeout` werden im Gegensatz zu Flash Lite 3.0 jetzt in Flash Lite 3.1 unterstützt. Die `setTimeout`-Funktion dient zum Ausführen einer bestimmten Funktion nach einer festgelegten Verzögerung (in Millisekunden). Bei der gemeinsamen Verwendung von `setTimeOut/clearTimeOut` und Inline-Eingabetext können Probleme für zeitkritische Anwendungen wie Onlinequizanwendungen oder Spiele entstehen, da bei der Eingabe von Text durch den Gerätebenutzer die gesamte ActionScript-Verarbeitung (einschließlich Timer) unterbrochen wird.

Streaming-Video-Suchfunktion

Flash Lite 3.1 unterstützt Suchfunktionen in gestreamten Flash-Videos (FLV). In Flash Lite 3.0 wurde die Suche bei der Videowiedergabe nur geringfügig unterstützt. Die Suche einer Position in Richtung Videoanfang oder Videoende in Relation zur aktuellen Kopfposition wurde nur bei kleinen HTTP-Videos unterstützt. Flash Lite 3.1 unterstützt jetzt auch die Suche in langen FLV-Videos. Die Suche in Richtung Videoanfang wird immer unterstützt, während für die Suche in Richtung Videoende einige Einschränkungen gelten. Die Suche in Richtung Videoende ist nur innerhalb des bereits heruntergeladenen Streams möglich. Beim Versuch, eine nach dem Ende des heruntergeladenen Streams liegende Position zu finden, wird keine Suche ausgeführt. Die Wiedergabe des Videos wird ab der aktuellen Kopfposition fortgesetzt. Die Suchfunktion wird gemeinsam mit der Netstream-Klasse verwendet. `Netstream.seek()` wird zur Suche in Richtung Videoanfang oder Videoende im HTTP-Videostream verwendet.

Streaming-mp3-Dateien

Flash Lite 3.1 unterstützt durch Verwendung des ActionScript-Audioobjekts das Streaming von MP3-Audiodateien. Bei Flash Lite 3.0 mussten MP3-Dateien vor dem Dekodieren und Abspielen vollständig heruntergeladen werden. Wenn das Gerät nicht genügend Speicher für die Verarbeitung der gesamten Datei zuweisen konnte, traten Fehler auf. Flash Lite 3.1 unterstützt progressives Audiostreaming über HTTP- und lokale Dateisystemprotokolle, wodurch MP3-Dateien von einem Webserver oder vom lokalen Dateispeicherort des Geräts wiedergegeben werden können. MP3-Daten werden beim Herunterladen verarbeitet. Dadurch wird der Speicherbedarf optimiert und Geräte können größere MP3-Dateien abspielen als dies bei Flash Lite 3.0 der Fall war.

ActionScript-Erweiterungen

Die Erweiterungsfunktion in Flash Lite 3.1 ermöglicht Drittentwicklern, Flash Lite-Funktionen und ActionScript-APIs mittels statischer oder dynamischer Bibliothek (DLL/Dynamically Linked Library) hinzuzufügen. Die DLL ermöglicht die Implementierung der Funktionen, und Flash Lite ermöglicht die Verwendung der DLL durch Aufrufen von ActionScript-APIs. Dies ermöglicht Flash Lite-Entwicklern die Interaktion mit APIs von Hostgeräten, indem sie eine DLL programmieren, die die Funktionen des Geräts mit den Funktionen des Flash-Players verbindet. Diese Erweiterungen können statisch oder dynamisch mit der Hostimplementierung verbunden werden. Entwickler können z.B. mithilfe dieser Funktion Anwendungen programmieren, die auf das Dateisystem des Geräts zugreifen oder die vom Gerät stammenden GPS-Daten nutzen und anderes mehr.

Hinweis: *ActionScript-Erweiterungsfunktion steht Drittentwicklern nur dann zur Verfügung, wenn sie vom Hersteller des Zielgeräts aktiviert wurde.*

Videoverbesserungen

Flash Lite 3.1 umfasst die folgenden Verbesserungen von Videofunktionen.

H.264-Videowiedergabe Flash Lite 3.1 kann, wie die Desktopversion von Flash Player 9.2, Flash 7-, 8- und 9-SWF-Dateien wiedergeben, die FLV verwenden. Hierzu wird auf dem Server (Flash Media Server oder HTTP) FLV durch H.264 ersetzt.

Videonachbearbeitung Flash Lite 3.1 unterstützt die Videonachbearbeitung (durch Verwendung der smoothing- und deblocking-Eigenschaften des Videoobjekts). Dadurch wird die Anzeigequalität von Videodateien auf Geräten verbessert.

Flash Lite 3.0-Funktionen

Neue Funktionen von Flash Lite 3.0

Leistungsverbesserungen Mit der aktuellen Version wird eine Reihe von Änderungen am Player eingeführt, die sowohl zu einer Verbesserung der Player-Leistung auf mobilen Geräten als auch zu einer Optimierung für den Benutzer des Geräts beim Anzeigen von und Interagieren mit Inhalten führt. Weitere Informationen finden Sie unter „Leistungsverbesserungen“ auf Seite 8.

Erweiterte Videofunktionen Unterstützt Flash-Video (FLV) und Gerätevideo. Weitere Informationen finden Sie unter „Erweiterte Videofunktionen“ auf Seite 8.

Durchsuchbarkeit von Webinhalten Unterstützt die Wiedergabe der meisten Flash 8-Inhalte auf Mobilgeräten mit geringen Qualitätseinbußen. Weitere Informationen finden Sie unter „Durchsuchbarkeit von Webinhalten“ auf Seite 8.

Lokale Dateisicherheit Bietet ein Sicherheitsschema für Flash Lite, das mit dem in Flash Player 8 eingeführten Sicherheitsmodell konsistent ist. Weitere Informationen finden Sie unter „Lokale Dateisicherheit“ auf Seite 9.

Inhaltsmanagement – Bietet eine API zum Lesen von SWF-Metadaten. Entwickler von Inhalten können mithilfe von Metadaten Inhalte auf Geräten segmentieren und bestimmte Aspekte der Verarbeitung von Inhalten steuern. Weitere Informationen hierzu finden Sie unter „Content Management“ auf Seite 10 .

Leistungsverbesserungen

Beschleunigung der Grafikwiedergabe Gibt Textbildläufe, Vektorübersetzungen und animierte Vektoren mit undurchsichtigen Füllfarben schnell und effizient wieder.

Verringerung des Laufzeitarbeitsspeichers Lädt ActionScript™-Klassen zur Laufzeit verlangsamt („lazy-loading“), sodass beim Starten nur die ActionScript-Klasse geladen wird, die zum Wiedergeben des ersten Bildes erforderlich ist. Andere ActionScript-Klassen werden dann vom Inhalt nach Bedarf geladen.

Erweiterte Videofunktionen

Flash Lite 2.x unterstützt nur Gerätevideo. Dies bedeutet, dass das Dekodieren und Wiedergeben von Video vom Gerät übernommen werden. Die Wiedergabe ist somit auf die von einem bestimmten Gerät unterstützten Videoformate eingeschränkt, Video kann nicht mit der Zeitleiste synchronisiert werden, die Lautstärke eines Videoclips kann nicht gesteuert werden und Gerätevideo kann nicht gedreht oder geneigt und nicht mit anderen Medien zusammengesetzt oder gemischt werden. (Das Video befindet sich immer im Vordergrund vor allen anderen Flash-Inhalten.)

Flash Lite 3.0 wurde um die Unterstützung für Flash Video (FLV) mithilfe von Versionen der On2- und Sorenson-Codecs, die für Mobilgeräte optimiert sind, erweitert. FLV-Video wird direkt im Flash Lite Player wiedergegeben, nicht vom Gerät. Auf diese Weise ist es irrelevant, ob Ihre Zielgeräte ein bestimmtes Videoformat unterstützen. Flash Lite 3.0 unterstützt die Wiedergabe von Flash Video mithilfe folgender Methoden:

- Einbetten von Video in SWF-Dateien
- Progressives Herunterladen von Video von lokalen Ordnern und Webservern (HTTP)
- Streaming von Video von einem Flash Media Server

  Vollständige Informationen zum Verwenden von FLV in Flash Lite 3.0 finden Sie unter „Mit Videos arbeiten“ auf Seite 79.

Durchsuchbarkeit von Webinhalten

Flash Lite 3.0 unterstützt die Wiedergabe der meisten Flash 8-Inhalte auf Mobilgeräten mit geringen Qualitätseinbußen. Aufgrund der hohen Anforderungen an Prozessor und Arbeitsspeicher, die für die Unterstützung von Flash 8 Expressiveness-Funktionen erforderlich sind, werden zahlreiche dieser Funktionen in Flash Lite 3.0 nicht unterstützt. Diese Expressiveness-Funktionen umfassen das Zwischenspeichern von Bitmaps, Füllmethoden, Bitmapfilter, erweiterte Striche, erweiterte Verläufe und 9-teilige Skalierung. Sobald Flash Lite eine dieser Funktionen in einer SWF-Datei erkennt, werden diese Objekte ohne die zusätzlichen Expressiveness-Merkmale wiedergegeben, um die Verwendbarkeit auf einem Mobilgerät zu gewährleisten.

In der folgenden Tabelle werden die Flash 8-Inhalte und -Funktionen beschrieben, die vollständig, teilweise oder gar nicht unterstützt werden.

Funktion	Unterstützung	Details
Grundlegende Flash 8-HTML-Tags	Ja	Der Flash Lite Player ist nun in der Lage, die meisten Flash 8-Inhalte zu erkennen und mit geringen Qualitätseinbußen wiederzugeben.
Sicherheitsverbesserungen	Ja	Das Sicherheitsmodell von Flash Lite ist nun mit dem Sicherheitsmodell des Desktop-Players konsistent. Weitere Einzelheiten finden Sie unter „Lokale Dateisicherheit“ auf Seite 9.
Flash Video	Ja	Der Flash Lite Player unterstützt nun Flash Video (FLV) mithilfe von Versionen der On2- und Sorenson-Codecs, die für Mobilgeräte optimiert sind. Darüber hinaus werden (Einwege-, keine Zweiwege-Echtzeit-) RTMP-Clientimplementierungen unterstützt. Weitere Einzelheiten finden Sie unter „Erweiterte Videofunktionen“ auf Seite 8.
Basis-HTML-Tags	Ja	Sorgt dafür, dass sich Flash Lite ähnlich wie Flash Player auf dem Desktop verhält.
FlashType-Text	Teilweise unterstützt	Die vollständige Unterstützung ist für die Anzeige von Inhalten nicht entscheidend. Text wird wiedergegeben, aber ohne vollständige FlashType-Implementierung.
Zwischenspeichern von Bitmaps, Effekte und Verbesserungen	Nein	Nicht implementiert, da für die meisten Funktionen Gleitkommaunterstützung erforderlich ist bzw. die Anforderungen an Prozessor- und Arbeitsspeicherressourcen übermäßig sind.
Fokussierte Verläufe	Nein	Nicht implementiert, weil sie sehr prozessorintensiv und für die Anzeige von Inhalten nicht wesentlich sind.

Lokale Dateisicherheit

In Flash Lite 2.x dürfen lokale SWF-Dateien mit anderen SWF-Dateien interagieren und daraus Daten laden, unabhängig davon, ob diese lokal oder dezentral sind. Dies erzeugt eine Sicherheitslücke, weil eine nicht vertrauenswürdige lokale SWF-Datei, die beispielsweise als E-Mail-Anlage heruntergeladen wurde, auf private Daten zugreifen kann, die auf dem Gerät gespeichert sind, und diese über eine HTTP-Verbindung zurück an den Autor des Inhalts senden kann.

Mit Flash Lite 3.0 wird ein neues Sicherheitsmodell eingeführt, das auf dem Sicherheitsmodell von Flash Player 8 basiert. Alle SWF-Dateien werden beim Laden in den Flash Lite Player in einer Sandbox platziert. SWF-Dateien werden entsprechend ihren Ursprungsorten verschiedenen Sandboxes zugewiesen.

SWF-Dateien aus dem Netzwerk werden immer in einer entfernten Sandbox platziert, die dem Domänenursprung entspricht. SWF-Netzwerkdateien können nur die Daten aus ihrer Ursprungsdomäne lesen. Auf Daten anderer Domänen können sie nur zugreifen, wenn diese Domänen der die Anforderung sendenden SWF-Datei eine explizite Berechtigung erteilen. Eine SWF-Datei aus einer a.com darf beispielsweise von dem Server in b.com lesen (beispielsweise mithilfe von `XML.load`), wenn b.com über eine Richtliniendatei verfügt, die den Zugriff von a.com aus (oder von allen Domänen) gestattet. Eine SWF-Datei aus a.com kann ein Cross-Scripting für eine SWF-Datei aus b.com (durch Aufrufen einer ActionScript™-Methode in der SWF-Datei in b.com) durchführen, wenn die SWF-Datei in b.com `System.security.allowDomain` aufruft, um den Zugriff von a.com aus zu gewähren.

SWF-Dateien mit lokalem Ursprung (lokales Dateisystem) werden wie folgt in einer von drei Sandboxes platziert.

Lokal mit Dateisystem SWF-Dateien können (beispielsweise mithilfe von `XML.load`) aus Dateien in lokalen Dateisystemen lesen, dürfen aber in keiner Weise mit dem Netzwerk kommunizieren.

Lokal vertrauenswürdig SWF-Dateien können lokale Dateien lesen, mit beliebigen Servern interagieren und alle anderen SWF-Dateien skripten. (In Flash Lite 2.1 befinden sich effektiv alle lokalen Dateien in dieser Sandbox.)

Lokal mit Netzwerk-Sandbox SWF-Dateien dürfen mit anderen Lokal-mit-Netzwerk-SWF-Dateien kommunizieren und Daten an Server senden (beispielsweise mithilfe von `XML.send`).

Hinweis: *Dieses neue Sicherheitsschema wirkt sich auf die Funktionalität beinahe aller ActionScript-Befehle aus, die das Laden von Daten oder Cross-Scripting beinhalten, einschließlich äußerst gängiger Funktionen wie* `getURL`*,* `loadMovie`*,* `loadMovieNum`*,* `loadVars`*,* `loadSound`*,* `XMLSocket.send` *und* `XMLSocket.load`*. Eine umfassende Liste aller API-Funktionalitäten, die von der Sicherheit betroffen sind, finden Sie auf der Flash Player Developer Center-Website unter „Flash Player 8 Security-Related APIs“ (PDF-Format).*

Informationen über die Arbeitsweise des Sicherheitsmodells von Flash 8 Player finden Sie in der Flash 8-Produkthilfe. Bei Verwendung dieser Hilfe ist zu beachten, dass folgende Unterschiede zwischen den Sicherheitsmodellen von Flash Lite 3.0 und Flash Player 8 bestehen:

- Die Verwaltung der Vertrauenswürdigkeit wird in Flash Lite von der Hostanwendung durchgeführt.
- In Flash Lite sind keine Vertrauenswürdigkeitskonfigurationsdateien erforderlich.
- Entwickler und Hersteller von SWF-Dateien sowie Netzbetreiber müssen alle Sicherheitseinstellungen in Flash Lite vornehmen. Es gibt weder benutzervermittelte Vertrauenswürdigkeit noch einen Einstellungen-Manager. Die Hostanwendung kann aber eventuell einen Mechanismus bereitstellen, mit dessen Hilfe Benutzer begrenzte Sicherheitseinstellungen vornehmen können. Beispielsweise kann der Host zulassen, dass Benutzer eine bestimmte SWF-Datei als vertrauenswürdig einstufen.
- Nicht zulässige Vorgänge schlagen ohne Anzeige einer Fehlermeldung in Flash Lite fehl.

Content Management

Flash Lite 3.0 stellt eine API zur Verfügung, mit deren Hilfe die Hostsoftware eines Geräts Informationen aus SWF-Metadaten lesen kann. Bei Metadaten handelt es sich um in SWF-Dateien eingebettete Standardinformationen. Wenn der Host diese Funktion implementiert, können Entwickler von Flash Lite-Inhalten mithilfe der Metadateninformationen Inhalte auf dem Gerät organisieren und kategorisieren.

Die folgenden Kategorien von Informationen können als Metadaten aufgenommen werden. Beachten Sie, dass einige dieser Kategorien möglicherweise abweichend (oder gar nicht) vom Hersteller der Geräte, für die Inhalte entwickelt werden, implementiert sind.

Titel	Beschreibung
Autor des Inhalts	Name des Inhaltsautors.
Firmenname	Name der Firma des Inhaltsautors.
Titel	Titel des Inhalts.
Beschreibung	Beschreibung des Inhalts.
Copyright-Informationen	Relevante Copyright-Details.
Inhaltsversion	Versionsnummer des Inhalts (NICHT die Flash-Version).
Inhaltstyp(en)	Kategorien für Inhaltstype(en) für die Inhaltsdatei. Eine Aufstellung der Inhaltstypen finden Sie in der folgenden Tabelle mit Inhaltstypen.
Weiterleitungssperre	Gibt an, ob die Inhaltsdatei auf ein anderes Gerät übertragen werden kann.
Unterstützte Anzeigeauflösungen	Liste der vom Inhalt unterstützten Auflösungen.

Titel	**Beschreibung**
Symbol	Typ und Position des Symbols für Inhalt.
Flash Player-Profil	Die Version von Flash/Flash Lite, für die der Inhalt veröffentlicht wurde.
Speichergröße	Die für den Inhalt erforderliche Speichergröße (Größe von `SharedObject`). Wenn die Größe nicht angegeben ist, wird dem Inhalt die Standardgröße zugewiesen, die von der Plattform festgelegt wird (im Allgemeinen 4 KB).
Hintergrund-Alpha	Bevorzugte/r Alpha-Wert und Füllmethode für die Hintergrundtransparenzfunktion.
Wiederholschleife	Wiederholschleifenverhalten: Inhalt wird entweder unendlich wiederholt oder nach dem letzten Bild beendet.
Standardqualität	Qualitätsstufe, mit der der Inhalt wiedergegeben werden soll.

Mithilfe der folgenden Inhaltstypen können Inhalte für Geräte segmentiert und bestimmte Aspekte der Verarbeitung von Inhalten gesteuert werden.

Inhaltstyp	**Beschreibung**
Bildschirmschoner	Flash-Animation, die angezeigt wird, wenn ein für das Gerät angegebenes Zeitlimit überschritten wird
Klingelton	Animation, die wiedergegeben wird, wenn das Telefon einen eingehenden Anruf empfängt
Hintergrund	Animiertes Hintergrundbild, das hinter dem Inaktivitäts- /Startbildschirm angezeigt wird
Spiel	Interaktives Spiel
Anwendung	Standalone-Anwendung
Präsentation	In einen Flash-Film konvertierte Präsentation
Symbol	Flash-basiertes animiertes Symbol
Film	Animierter Flash-Film
Skin	Skin oder Motiv für das Telefon oder eine einzelne Anwendung
Start	Animation beim Starten
Herunterfahren	Animation beim Herunterfahren
Andere	Flash-Inhalte, die zu keiner der oben genannten Kategorien gehören.

Flash Lite 2.x-Funktionen

Funktionen von Flash Lite 2.x

Flash Lite 2.0 und 2.1 (gemeinsam als 2.x bezeichnet) basieren auf Flash Player7 und unterstützen die meisten in dieser Version von Flash Player verfügbaren Funktionen, darunter die XML-Verarbeitung und ActionScript 2.0. Flash Lite 2.x bietet außerdem verschiedene neue Funktionen, die in Flash Player 7 nicht verfügbar sind und speziell für Mobilanwendungen entwickelt wurden.

Neue Funktionen von Flash Lite 2.1

Flash Lite 2.1 bietet jetzt Unterstützung für die Inline-Texteingabe. In früheren Versionen von Flash Lite nutzten die Eingabetextfelder das generische Texteingabe-Dialogfeld des Gerätes. In Flash Lite 2.1 können Eingabetextfelder direkt bearbeitet werden.

Hinweis: *Geräte, die komplexe Sprachen wie z. B. Hebräisch, Arabisch, Farsi und einige asiatische Sprachen unterstützen, können leider nicht für die Inline-Texteingabe verwendet werden. Bei diesen Geräten weisen die Eingabetextfelder unter Flash Lite 2.1 die gleichen Funktionen wie unter Flash Lite 2.0 auf.*

Flash Lite 2.1 bietet die Socketkommunikation über die XMLSocket-Klasse auf Geräten, die diese Funktion unterstützen. (Weitere Informationen finden Sie im *Flash Lite 2.x und 3.0 ActionScript-Referenzhandbuch*.) Weitere Informationen zu Eingabetextfeldern finden Sie unter „Text und Schriftarten verwenden“ auf Seite 54.

Flash Lite 2.1 fügt darüber hinaus Werkzeuge zur Entwicklung von Flash Lite-Anwendungen hinzu, die auf Geräten ausgeführt werden, die die Plattform „Binary Runtime Environment for Wireless®“ (BREW®) von QUALCOMM Incorporated verwenden. Weitere Informationen zu diesen Werkzeugen finden Sie unter „Flash Lite-Anwendungen für BREW entwickeln“ auf Seite 95.

Neue Funktionen von Flash Lite 2.0

Flash Lite 2.0 umfasst die folgenden neuen Funktionen:

- Unterstützung von ActionScript 2.0 und damit verbundene erweiterte Programmierungstechniken mit Klassen, Schnittstellen, strikter Datentypisierung usw.
- Gerätevideo-Wiedergabe
- Lokale, permanente Datenspeicherung (gemeinsame Flash Lite-Objekte)
- Unterstützung des Ladens gerätespezifischer Sound- und Bildformate
- Informationen zu neuen Systemfähigkeiten
- Unterstützung zusätzlicher Gerätetasten, einschließlich Unterstützung von QWERTY-Tastaturen sowie Unterstützung von bis zu 11 Softtasten
- Rich-Text-Formatierung (Teilunterstützung)
- Möglichkeit zum Steuern der Hintergrundbeleuchtungsdauer und zum Festlegen benutzerdefinierter Fokusrechteck-Farben
- Synchronisierter Gerätesound
- Unterstützung von XML-Verarbeitung

  Die folgenden Funktionen von Flash Player 7 sind nicht in Flash Lite 2.0 verfügbar:

- Mehrere ActionScript-Klassen, die in Flash Player 7 verfügbar sind, werden in Flash Lite 2.0 nicht oder nur teilweise unterstützt. Weitere Informationen zu verfügbaren Funktionen von ActionScript finden Sie im Handbuch *Einführung in Flash Lite 2.0 ActionScript*.
- Socketkommunikation über die XMLSocket-Klasse (verfügbar in Flash Lite 2.1)
- Unterstützung der Kommunikation mit Flash Media Server (verfügbar in Flash Lite 3.0)
- Entfernte gemeinsame Objekte (Teilunterstützung von lokalen gemeinsamen Objekten)
- Native Unterstützung der Wiedergabe von Flash Video (FLV) (verfügbar in Flash Lite 3.0)
- Unterstützung von Flash Application Protocol (dem von Flash Remoting verwendeten Binärdatenkommunikationsprotokoll)

- CSS-Formatierung (Cascading Style Sheet) mit Textfeldern
- Masken mit Geräteschriftarten
- Bitmapglättung bei der Wiedergabe in hoher Qualität

Flash Lite 2.0 ActionScript

Flash Lite 2.0 ActionScript ist die Skriptsprache, die in Flash Lite 2.0- und 2.1-Anwendungen verwendet wird. Sie enthält einige der Funktionen von ActionScript in Flash Player 7, jedoch nicht alle. Flash Lite 2.0 enthält verschiedene ActionScript-Zusätze und -Erweiterungen, mit denen Sie beispielsweise Informationen über das Gerät abrufen, Telefonanrufe durchführen oder die Dauer der Hintergrundbeleuchtung steuern können.

Beim Entwickeln von Anwendungen für Flash Lite 2.0 können Sie die Syntax von ActionScript 2.0 oder von ActionScript 1.0 verwenden. ActionScript 2.0 bietet Authoring-Unterstützung für Klassen, Schnittstellen und strikte Datentypisierung. Mit der ActionScript 2.0-Syntax stellt der ActionScript-Compiler bessere Debugging-Informationen bereit. Zudem wird ein besseres Programmdesign gefördert.

Weitere Informationen zum Lernen von Flash Lite 2.0 ActionScript finden Sie in den folgenden Büchern und Themen:

- *Einführung in Flash Lite 2.x und 3.0 ActionScript*
- *Flash Lite 2.x und 3.0 ActionScript-Referenzhandbuch*
- „ActionScript 2.0 in Flash – Arbeitshandbuch“ in *Flash verwenden*

Gerätevideo-Wiedergabe

Flash Lite 2.0 kann Videos in jedem Format wiedergeben, das von dem Zielgerät nativ unterstützt wird. Beispielsweise werden Videos auf einigen Geräten im 3GP-Videoformat aufgezeichnet und wiedergegeben. Andere unterstützen AVI- oder MPEG-Videos. Bei der Wiedergabe übergibt Flash Lite die ursprünglichen Videodaten an das Gerät, auf dem die Daten direkt dekodiert und auf dem Bildschirm dargestellt werden. Sie haben folgende Möglichkeiten, um Videos in Anwendungen zu integrieren:

- Speichern der ursprünglichen Videodaten als Bundle in der SWF-Datei.
- Laden einer externen Videodatei aus der Speicherkarte des Geräts oder über das Netzwerk.

Zum Steuern der Videowiedergabe in Flash Lite 2.0 wird das Video-Objekt von ActionScript verwendet. Das Video-Objekt in Flash Lite 2.0, das zuerst in Flash Player 6 verfügbar war, weist zusätzliche Methoden zum Steuern der Videowiedergabe auf, z. B. die Methoden `Video.play()` und `Video.pause()`. Sie können auch mit dem Array `System.capabilities.videoMIMETypes` ermitteln, welche Videoformate ein Gerät unterstützt.

Weitere Informationen zum Video-Objekt und zur Verwendung von Video in Flash Lite finden Sie unter „Mit Videos arbeiten“ auf Seite 79.

Laden von gerätespezifischen Sound- und Bildformaten

In Flash Lite 2.0 können Sie jede Bild- oder Sounddatei laden, die in einem vom Gerät unterstützten Format vorliegt. Um externe Bilder zu laden, verwenden Sie die globale Funktion `loadMovie()` oder die Methode `MovieClip.loadMovie()`. Wenn das Gerät beispielsweise das Dateiformat PNG unterstützt, können Sie mit dem folgenden Code eine PNG-Datei von einem Webserver in die Movieclip-Instanz image_mc laden:

```
image_mc.loadMovie("http://www.adobe.com/images/mobile.png");
```

Um externe Sounds zu laden, verwenden Sie die Methode `Sound.loadSound()`. In Flash Lite 2.0 können Sie mit dieser Methode jedes Soundformat laden, das vom Gerät unterstützt wird (z. B. MIDI oder SMAF). Externe Gerätesounds müssen vor der Wiedergabe vollständig in den Arbeitsspeicher geladen werden.

Wie in Flash Lite 1.x können Sie auch in Flash Lite 2.0 Gerätesounds wiedergeben, die in der SWF-Datei integriert sind. Weitere Informationen finden Sie unter „Gerätesound-Bundle verwenden“ auf Seite 69.

Weitere Informationen zum Laden von externen Bildern und Sound finden Sie in folgendem Thema:

Siehe auch

„Externe Gerätesounds wiedergeben“ auf Seite 75

Gemeinsame Flash Lite-Objekte

Mit gemeinsamen Flash Lite-Objekten können Sie Daten permanent auf dem Gerät des Benutzers speichern. Beispielsweise können Sie ein gemeinsames Objekt verwenden, um Informationen zwischen mehreren Anwendungssitzungen zu speichern, z. B. Benutzereinstellungen oder Punktestände in Spielen. Gemeinsame Flash Lite-Objekte werden über die Klasse SharedObject gelesen und geschrieben. Weitere Informationen zum Verwenden von gemeinsamen Flash Lite-Objekten finden Sie unter der Klasse `SharedObject` im *Flash Lite 2.x und 3.0 ActionScript-Referenzhandbuch*.

Hinweis: *In der Flash Lite 2.0-Implementierung von gemeinsamen Objekten ist es nicht möglich, dieselben Daten in mehreren SWF-Dateien gemeinsam zu verwenden. Außerdem unterstützt keine der Flash Lite-Versionen entfernte Objekte, die gemeinsam mit Flash Media Server verwendet werden.*

Synchronisierter Gerätesound

In bisherigen Versionen von Flash Lite konnten Sie nur native Flash-Sounds mit Animationen in der Zeitleiste synchronisieren. Dies war mit Gerätesounds nicht möglich, die direkt über das Gerät und nicht nativ über Flash Lite wiedergegeben werden. In Flash Lite 2.0 können Sie Gerätesounds mithilfe der neuen Eigenschaft `_forceframerate` mit der Zeitleiste synchronisieren. Wenn diese Eigenschaft auf `true` gesetzt wird, verwirft Flash Lite gegebenenfalls Einzelbilder aus der Animation, um die in der SWF-Datei angegebene Bildrate aufrecht zu erhalten. Weitere Informationen finden Sie unter „Gerätesounds mit Animationen synchronisieren“ auf Seite 75 und der `forceframerate-Eigenschaft` im *Flash Lite 2.x- und 3.0 ActionScript-Referenzhandbuch*.

Textfunktionen

Die folgenden auf die Verarbeitung von Text bezogenen Funktionen sind neu in Flash Lite 2.0:

- Sämtliche Textdarstellungen in Flash Lite 2.0 basieren auf Unicode.
- Flash Lite 2.0 bietet Teilunterstützung von HTML-Formatierung und der ActionScript-Klasse TextFormat.

  Weitere Informationen zu Textfeldern in Flash Lite 2.x finden Sie unter „Text und Schriftarten verwenden“ auf Seite 54.

Zusätzliche Unterstützung von Tasten

Flash Lite 2.0 bietet zusätzliche Unterstützung von Gerätetasten, darunter QWERTY-Tastaturen sowie bis zu 12 Softtasten (einschließlich der linken und der rechten Standard-Softtaste).

Kapitel 2: Erste Schritte mit Flash® Lite

Flash Lite-Anwendung „Hello World“

In diesem einfachen Tutorial lernen Sie die Authoring- und Testfunktionen von Adobe Flash CS4 Professional für Mobilgeräte kennen. Außerdem werden Sie mit dem allgemeinen Arbeitsablauf vertraut gemacht, der beim Erstellen von Inhalten mit Adobe Flash Lite zu beachten ist. In diesem Abschnitt erstellen Sie eine einfache Flash Lite-Anwendung und testen sie im Adobe Device Central CS4-Emulator. Eine umfangreichere Beispielanwendung finden Sie in „Tutorial: Flash Lite-Anwendungen erstellen“ auf Seite 19.

Im Rahmen dieses Tutorials entwickeln Sie Inhalte für den eigenständigen Flash Lite 2.0 Player. Das Verfahren ist für ein generisches Gerät vorgesehen.

Entscheiden Sie zunächst, für welche Geräte und für welchen Flash Lite-Inhaltstyp die Anwendung vorgesehen ist.

Eine einfache Flash Lite-Anwendung konfigurieren und erstellen

1 Starten Sie Flash.

2 Wählen Sie auf dem Flash-Hauptbildschirm Neu erstellen > Flash-Datei (Mobil). Adobe Device Central wird von Flash geöffnet und die Registerkarte „Neues Dokument“ wird angezeigt.

3 Wählen Sie auf der Registerkarte „Neues Dokument“ im Feld „Player-Version“ die Option FlashLite 2.0, im Feld „ActionScript-Version“ die Option „Adobe ActionScript™ 2.0“ und im Feld „Inhaltstyp“ die Option „Standalone-Player“.

4 Klicken Sie am unteren Bildschirmrand auf „Benutzerdefinierte Größe für alle ausgewählten Geräte“. Damit sind Sie in der Lage, Inhalte für den eigenständigen Flash Lite Player zu erstellen.

5 Klicken Sie auf „Erstellen“. Daraufhin kehren Sie zu Flash zurück und ein neues Dokument wird erstellt und zwar mit vorgegebenen Veröffentlichungseinstellungen und (wenn Sie ein Gerät angeben) mit der richtigen Größe für das ausgewählte Gerät.

6 Wählen Sie im Bedienfeld „Werkzeuge“ das Textwerkzeug aus und erstellen Sie auf der Bühne ein Textfeld.

   Geben Sie **Hello, World!** (oder anderen Text) in das Textfeld ein.

7 Wählen Sie Steuerung > Film testen, um die Anwendung an Adobe Device Central zu exportieren und um sie im Adobe Device Central-Emulator anzuzeigen.

   Hinweis: *Beim Testen in Device Central können Sie den Geräte- und Inhaltstyp ändern, um Ihre Anwendung auf einer anderen Plattform anzuzeigen. Doppelklicken Sie zu diesem Zweck im Bedienfeld „Bibliothek“ auf ein Gerät und wählen Sie im Bedienfeld „Inhaltstyp“ einen neuen Inhaltstyp aus. Bei der Rückkehr zu Flash werden die zuletzt im Emulator verwendeten Einstellungen von Flash gespeichert.*

8 Um zu Flash zurückzukehren, wählen Sie „Datei“ > „Zurück zu Flash“ aus.

Emulator und Gerätedatenbank von Adobe Device Central

Mit dem Adobe Device Central-Emulator können Sie testen, wie Ihre Inhalte auf dem eigentlichen Gerät aussehen und ausgeführt werden. Sie können ein anderes Testgerät oder einen anderen Inhaltstyp auswählen und detaillierte Informationen zur Anwendung anzeigen.

Device Central enthält eine umfangreiche Datenbank mit Mobilgeräten, aus der Sie eine Auswahl treffen und Ihre eigene Testumgebung erstellen können. Weitere Informationen hierzu finden Sie in der Device Central-Dokumentation.

Arbeitsablauf für die Entwicklung von Flash Lite-Anwendungen

Die Entwicklung von Flash Lite-Inhalten geht immer nach dem gleichen Schema vonstatten und umfasst die folgenden Schritte:

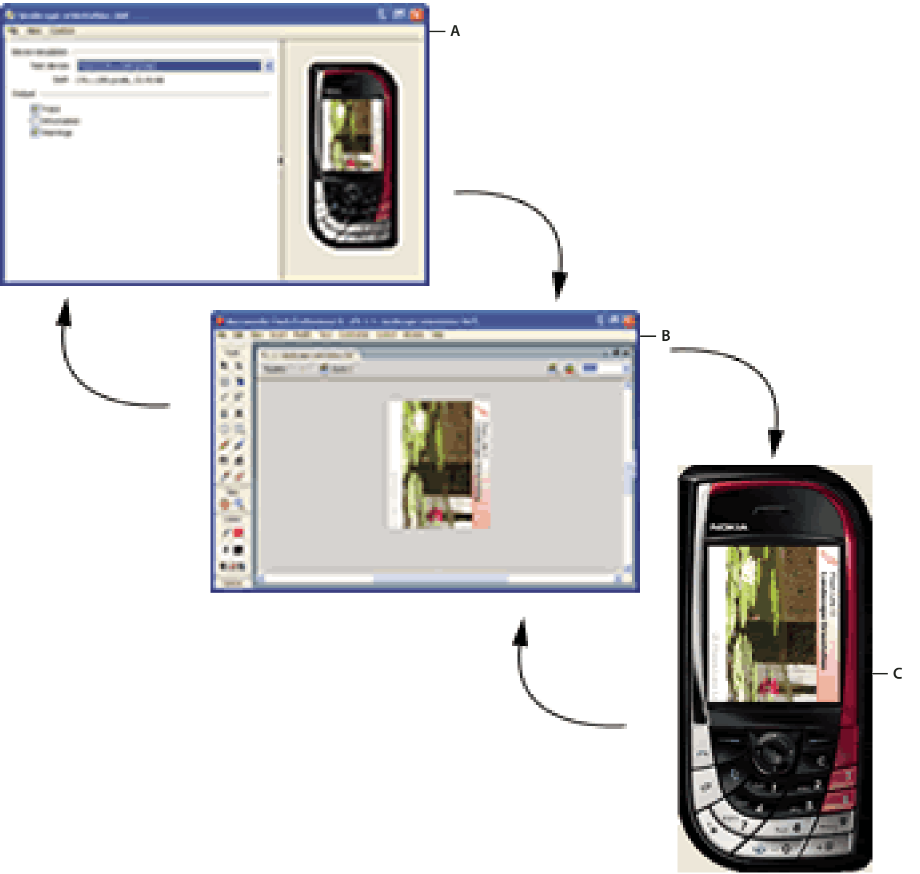
Zielgeräte und Flash Lite-Inhaltstyp identifizieren Verschiedene Geräte haben u. a. unterschiedliche Bildschirmgrößen, unterstützen verschiedene Audioformate und haben verschiedene Bildschirmfarbtiefen. Diese Faktoren können sich auf das Design und die Implementierung einer Anwendung auswirken.

Außerdem unterstützen nicht alle Geräte die gleichen Flash Lite-Inhaltstypen, z. B. Bildschirmschoner, eigenständige Anwendungen oder animierte Klingeltöne. Über den Inhaltstyp, für den Sie eine Anwendung entwickeln, wird darüber hinaus festgelegt, welche Funktionen für die Anwendung verfügbar sind. Weitere Informationen finden Sie unter „Flash Lite-Inhaltstypen“ auf Seite 17.

Erstellen Sie Ihre eigene Anwendung in Flash und testen Sie diese in Adobe Device Central Der Lieferumfang von Adobe Flash Professional umfasst einen Adobe Device Central CS4-Emulator, mit dem Sie Ihre Anwendung testen können, ohne sie auf ein Gerät übertragen zu müssen. Mithilfe des Emulators können Sie das Anwendungsdesign optimieren und sämtliche Probleme beheben, bevor Sie die Anwendung auf einem Mobilgerät testen.

Anwendung auf dem bzw. den Zielgerät(en) testen Dieser Schritt ist wichtig, da der Adobe Device Central-Emulator einige Aspekte des Zielgeräts wie z. B. die Prozessorgeschwindigkeit, die Farbtiefe oder die Netzwerklatenz nicht imitieren kann. Eine Animation, die im Emulator problemlos ausgeführt werden kann, wird unter Umständen auf dem Gerät aufgrund einer geringeren Prozessorgeschwindigkeit langsamer wiedergegeben. Farbverläufe, die im Emulator glatte Übergänge aufweisen, ergeben auf dem Gerät möglicherweise Streifen. Nachdem Sie die Anwendung auf einem Mobilgerät getestet haben, müssen Sie das Design möglicherweise im Flash-Authoring-Tool überarbeiten.

Die folgende Abbildung veranschaulicht die bei der Entwicklung und beim Testen erforderlichen Schritte:



***A.** FLA-Datei in Adobe Flash bearbeiten* ***B.** Testen im Adobe Device Central-Emulator* ***C.** Testen auf einem Gerät*

Flash Lite-Inhaltstypen

Bevor Sie mit der Entwicklung einer Flash Lite-Anwendung beginnen, müssen Ihnen folgende Informationen vorliegen:

- Das Gerät bzw. die Geräte, auf dem bzw. denen der Inhalt ausgeführt wird. Diese Geräte werden auch als *Zielgeräte* bezeichnet. Der Flash Lite Player ist auf einer Vielzahl von Geräten installiert. Eine Liste der Geräte, auf denen Flash Lite installiert ist, finden Sie auf der Adobe-Website auf der Seite www.adobe.com/go/mobile_supported_devices_de/.
- Die von den Zielgeräten unterstützten Flash Lite-Inhaltstypen. Jede Flash Lite-Installation unterstützt einen oder mehrere Anwendungsmodi, die auch als *Inhaltstypen* bezeichnet werden. Auf einigen Geräten wird Flash Lite z. B. für Flash-basierte Bildschirmschoner oder animierte Klingeltöne verwendet. Auf anderen Geräten dient Flash Lite zur Wiedergabe von Flash-Inhalten, die in Webseiten für Mobilgeräte eingebettet sind. Nicht alle Inhaltstypen unterstützen sämtliche Flash Lite-Funktionen.

Welche Flash Lite-Funktionen für Ihre Anwendung zur Verfügung stehen, hängt vom jeweils ausgewählten Gerät und Inhaltstyp ab. Beispielsweise können mit einer Flash-Anwendung, die als Bildschirmschoner ausgeführt wird, in der Regel keine Netzwerkverbindungen hergestellt oder keine Daten heruntergeladen werden.

Mit den Flash Lite-Testfunktionen in Adobe Device Central können Sie testen, wie Anwendungen auf verschiedenen Geräten und bei unterschiedlichen Flash Lite-Inhaltstypen angezeigt und ausgeführt werden. Auf diese Weise können Sie feststellen, ob eine Anwendung Funktionen verwendet, die für den von Ihnen entwickelten Inhaltstyp nicht verfügbar sind. Weitere Informationen zur Auswahl von Zielgeräten und Inhaltstypen finden Sie unter „Flash Lite-Inhalte testen“ auf Seite 118 in *Flash Lite 2.x- und 3-Anwendungen entwickeln*.

Kapitel 3: Tutorial: Flash Lite-Anwendungen erstellen

In diesem Tutorial erstellen Sie eine Adobe Flash Lite-Anwendung, die für ein fiktives Restaurant namens Café Townsend Werbung macht. Benutzer können sich die Tagesgerichte des Restaurants oder ein Video über den Küchenchef ansehen und telefonisch einen Tisch reservieren. Benutzer geben beim Reservieren auch ihr bevorzugtes Lokal an. Diese Auswahl wird mithilfe eines gemeinsamen Objekts gespeichert und ist in der jeweils folgenden Sitzung verfügbar.

Fertig gestellte Anwendung anzeigen

Die vollständige Anwendung ist in zwei Versionen verfügbar: eine Version verwendet eine 3GP-Gerätevideodatei und die andere Version verwendet eine FLV-Datei. Beide Versionen können mit dem Adobe Device Central-Emulator angezeigt werden. Außerdem können Sie die vollständige Anwendung auf einen eigenständigen Flash Lite Player übertragen, der auf einem Mobilgerät installiert ist.

Device Central unterstützt nicht alle Adobe ActionScript-Befehle. Die Befehle `fscommand2("Quit")` oder `getURL()` werden beispielsweise in Device Central nicht unterstützt. Device Central sendet Meldungen an das Ausgabefenster, um zu bestätigen, dass die Befehle aufgerufen wurden. Wählen Sie zum Anzeigen der Meldungen „Fenster“ > „Flash-Ausgabe“ (Device Central CS4) oder „Ansicht“ > „Flash-Ausgabe“ > „Anzeigen“ (Device Central CS3). Weitere Informationen finden Sie unter „Vom Emulator nicht unterstützte Flash Lite-Funktionen“ auf Seite 124.

1 Laden Sie die Tutorial-Dateien herunter und dekomprimieren Sie sie. Wechseln Sie zu www.adobe.com/go/learn_flt_samples_and_tutorials_de. Laden Sie die komprimierte Datei für Ihre Flash Lite-Version herunter.

2 Wählen Sie in Flash „Datei“ > „Öffnen“ und suchen Sie im Tutorial-Ordner nach der Datei cafe_tutorial_complete.fla oder cafe_tutorial_FLV_complete.fla.

3 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung in Device Central zu testen.

4 Gehen Sie folgendermaßen vor, um die Anwendung auszuprobieren:

- Wählen Sie im Startbildschirm den Menüeintrag „Specials“ aus, indem Sie auf die Nach-unten-Taste klicken. Klicken Sie auf die Auswahltaste, um den Tagesgerichtbildschirm anzuzeigen.
- Klicken Sie in diesem Tagesgerichtbildschirm auf die rechte Softtaste („Next“) des Emulators, um jeweils ein Bild und eine Beschreibung zum jeweiligen Tagesgericht anzuzeigen. Klicken Sie auf die linke Softtaste („Home“), um zum Hauptbildschirm zurückzukehren.
- Klicken Sie auf den Menüeintrag „View Video“, um das Video anzuzeigen. Klicken Sie auf die linke Softtaste („Home“), um zum Hauptbildschirm zurückzukehren. Sie können auch auf die rechte Softtaste (Pause) klicken, um das Video anzuhalten.
- Klicken Sie im Startbildschirm auf die linke Softtaste („Set Location“), um zu dem Bildschirm zu wechseln, in dem Sie das gewünschte Lokal auswählen können. Wählen Sie ein bevorzugtes Lokal für Ihre Reservierungen aus und klicken Sie dann auf die linke Softtaste („Save“), um den Standort festzulegen und zum Startbildschirm zurückzukehren (oder klicken Sie auf „Cancel“).

- Klicken Sie im Startbildschirm auf den Menüeintrag „Reservations“, um beim entsprechenden Restaurant anzurufen. Bestätigen Sie im Ausgabefenster, dass die Telefonnummer mit dem im Standort-Bildschirm ausgewählten Ort übereinstimmt.

Anwendung erstellen

Menü für den Startbildschirm erstellen

1 Wählen Sie in Flash „Datei“ > „Öffnen“ und suchen Sie im Tutorial-Ordner nach der Datei cafe_tutorial_start.fla.

2 Wählen Sie „Datei“ > „Speichern unter“ und speichern Sie die Datei als cafe_tutorial.fla.

3 Wählen Sie in der Zeitleiste (Fenster > Zeitleiste) Bild 1 in der Ebene „Main menu“ aus.

4 Öffnen Sie das Bedienfeld „Bibliothek“ („Fenster“ > „Bibliothek“) und ziehen Sie eine Instanz der Schaltflächen „Specials“ auf die Bühne. Führen Sie eine der folgenden Aktionen aus:

   a Platzieren Sie die Schaltfläche unter dem Textfeld, das Informationen zum Restaurant enthält.

   b Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ die Bezeichnung **specials_btn** ein, während die Schaltfläche markiert ist.

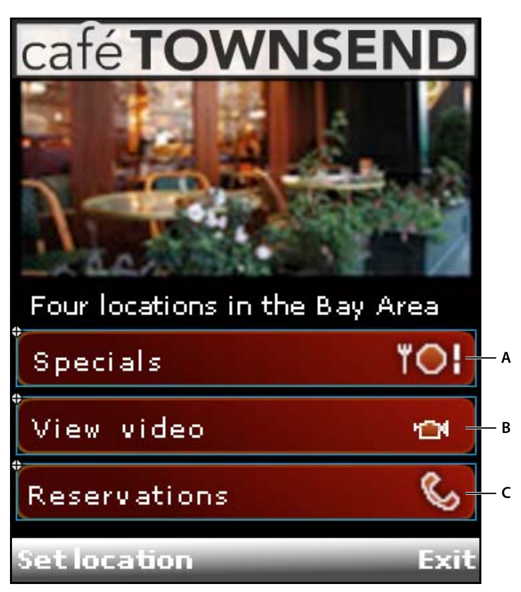
5 Ziehen Sie eine Instanz der video-Schaltfläche aus dem Bedienfeld „Bibliothek“ auf die Bühne. Führen Sie eine der folgenden Aktionen aus:

   - Platzieren Sie die video-Schaltfläche unter der Schaltfläche „Specials“.
   - Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ die Bezeichnung **video_btn** ein, während die Schaltfläche markiert ist.

6 Ziehen Sie eine Instanz der Schaltfläche „Reservations“ aus dem Bedienfeld „Bibliothek“ auf die Bühne. Führen Sie eine der folgenden Aktionen aus:

   - Platzieren Sie die Reservations-Schaltfläche unter der video-Schaltfläche.
   - Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ die Bezeichnung **reservations_btn** ein, während die Schaltfläche „Reservations“ markiert ist.

7 Nun sollte die Bühne ungefähr folgendermaßen aussehen:



A. *specials_btn* ***B.*** *video_btn* ***C.*** *reservations_btn*

8 Wählen Sie in der Zeitleiste Bild 1 der Ebene ActionScript aus. Öffnen Sie das Bedienfeld „Aktionen“ (Fenster > Aktionen) und geben Sie folgenden Code ein:

```
stop();
_focusrect = false;
fscommand2("SetSoftKeys", "Set Location", "Exit");
fscommand2("Fullscreen", "true");
```

Dieser Code bewirkt Folgendes:

- Der Abspielkopf wird bei diesem Bild angehalten.
- Deaktiviert das Fokusrechteck, mit dem Flash Lite Schaltflächen oder Textfelder, die den Fokus besitzen, standardmäßig umgibt (siehe „Das Fokusrechteck“ auf Seite 40).
- Die in der Anwendung verwendeten Softkeys werden registriert.
- Zeigt die Anwendung im Vollbildmodus an.

Wenn Flash Lite im Vollbildmodus ausgeführt wird, sind die mit dem Befehl `SetSoftKeys` festgelegten Bezeichnungen nicht sichtbar. Aus diesem Grund müssen Sie der Bühne benutzerdefinierte Softkeybezeichnungen hinzufügen.

9 Fügen Sie den folgenden Code auf Bild 1 ein, um Schaltflächenereignisse für die Menüschaltflächen zu verarbeiten und den Fokus auszuwählen:

```
// Set initial focus when the application
// starts and also upon returning to the main
// screen from another screen.
if (selectedItem == null) {
    Selection.setFocus (specials_btn);
} else {
    Selection.setFocus (selectedItem);
}
// Assign onPress event handlers to each menu button,
// and set selectedItem variable to selected button
// object:
specials_btn.onPress = function () {
    gotoAndStop ("specials");
    selectedItem = this;
};
video_btn.onPress = function () {
    gotoAndStop ("video");
    selectedItem = this;
};
reservations_btn.onPress = function () {
    if (location_so.data.phoneNumber == undefined) {
    // User hasn't specified location so
    // go to "set location" screen:
        gotoAndStop ("locations");
    }
    else {
        // Call number in shared object:
        var phoneNum = location_so.data.phoneNumber;
        getURL ("tel:" + phoneNum);
    }
    selectedItem = this;
};
```

Die den Schaltflächen `specials_btn` und `video_btn` zugeordneten `onPress`-Ereignisprozeduren setzen den Abspielkopf auf Bilder mit der Bezeichnung „specials" bzw. „video".

Wenn Benutzer die Option „Reservations" auswählen, wählt die `onPress`-Ereignisprozedur die im gemeinsamen Objekt `location_so` festgelegte Telefonnummer an. (Den Code zum Anlegen des gemeinsamen Objekts erstellen Sie später in diesem Tutorial.) Wenn ein Benutzer noch keinen Standort für Reservierungsanrufe festgelegt hat, setzt die Anwendung den Abspielkopf auf das Bild „locations" . Dort kann dann das bevorzugte Lokal für Reservierungen ausgewählt werden.

10 Fügen Sie nun Bild 1 den folgenden Code hinzu, um einen Tasten-Listener für die linke und rechte Softtaste zu erstellen:

```
Key.removeListener(myListener);
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    var keyCode = Key.getCode();
    if (keyCode == ExtendedKey.SOFT1) {
        // Handle left soft key event:
        gotoAndStop("locations");
    } else if (keyCode == ExtendedKey.SOFT2) {
        // Handle right soft key event:
        fscommand2("Quit");
    }
};
Key.addListener(myListener);
```

In diesem Code wird ein Tasten-Listener-Objekt verwendet, um die Ereignisse der rechten und linken Softtaste zu verarbeiten. Drückt der Benutzer die linke Softtaste, springt der Abspielkopf zum Bild „locations". Die rechte Softtaste schließt die Anwendung.

Weitere Informationen zur Verwendung von Event-Listenern finden Sie unter „Tastendruck-Ereignisse mithilfe eines Tasten-Listeners verarbeiten" auf Seite 50.

11 Fügen Sie Code hinzu, um das gemeinsame Objekt zu initialisieren, in dem die Auswahl des für Reservierungen bevorzugten Lokals gespeichert wird:

```
// Define Shared Object listener function:
function so_listener (the_so:SharedObject) {
    if (the_so.getSize () == 0) {
    // The shared object doesn't exist, so the user
    // hasn't set a preference yet.
    }
    SharedObject.removeListener ("location");
}
// Create shared object:
location_so = SharedObject.getLocal ("location");
// Add SharedObject listener object:
SharedObject.addListener ("location", this, "so_listener");
```

12 Wählen Sie „Datei" > „Speichern".

13 Wählen Sie zum Testen Ihrer Arbeit in Device Central „Steuerung" > „Film testen" aus.

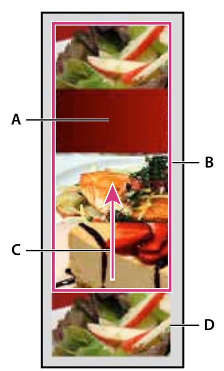
Sie können jetzt durch den Wechsel des Fokus auf die entsprechende Schaltfläche einen Menüeintrag auswählen und dann auf die Auswahltaste klicken (bzw. die Eingabetaste auf der Computertastatur drücken). Öffnen Sie das Fenster „Ausgabe" zur Anzeige von nicht unterstützten Befehlen.

Tagesgerichtbildschirm erstellen

Erstellen Sie die Animation, die für einen fließenden Übergang zwischen den Bildern der Tagesgerichte sorgt. Fügen Sie Benutzeroberflächenelemente und ActionScript hinzu, mit denen der Anwender zwischen den einzelnen Bildern navigieren kann. Fügen Sie ActionScript hinzu, um den Namen und die Beschreibung für jedes Tagesgericht unter dem Bild anzuzeigen.

Bildanimation erstellen

Verwenden Sie zum Erstellen der Animation einen bereits zusammengestellten Movieclip, in dem die Bilder aller Tagesgerichte in einer vertikalen Spalte angeordnet sind. Mit einer Maskenebene gewährleisten Sie, dass nur jeweils eines der Bilder sichtbar ist. Eine Tween-Animation bewegt die Bilder durch die Maske. Das letzte Bild des Movieclips ist ein Duplikat des ersten Bildes, damit die Animationssequenz zum Ausgangspunkt zurückkehren kann, nachdem sich der Benutzer das letzte Bild angesehen hat.



A. Maskenebenenobjekt *B. Maskierter Movieclip mit Bildern* *C. Richtung des Tweens* *D. Dupliziertes Bild*

1 Öffnen Sie die Datei cafe_tutorial.fla, die Sie im Abschnitt „Menü für den Startbildschirm erstellen“ auf Seite 20 erstellt haben.

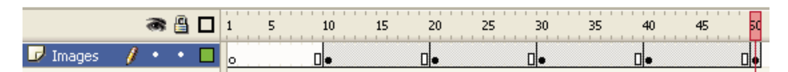
2 Wählen Sie in der Zeitleiste in der Ebene mit der Bezeichnung „Specials Photos“ das Schlüsselbild in Bild 10 aus.

3 Öffnen Sie das Bedienfeld „Bibliothek“ und ziehen Sie das Symbol für den Movieclip „Specials“ auf die Bühne.

4 Achten Sie darauf, dass der Movieclip „Specials“ noch ausgewählt ist. Stellen Sie dann die *x*- und die *y*-Koordinate im Eigenschafteninspektor auf **0** ein.

   Dadurch wird die linke obere Ecke des Movieclips „Specials“ an der linken oberen Ecke der Bühne ausgerichtet.

5 Wählen Sie in der „Specials Photos“-Ebene die Bilder 20, 30, 40 und 50 aus, und drücken Sie die Taste F6, um die Schlüsselbilder einzufügen. Wählen Sie Bild 51 und drücken Sie F7, um ein leeres Schlüsselbild einzufügen. Nun sollte die Zeitleiste wie in der folgenden Abbildung aussehen:



6 Wählen Sie das Schlüsselbild in Bild 20 aus. Wählen Sie dann den „Specials“-Movieclip aus und weisen Sie der *y*-Koordinate im Eigenschafteninspektor den Wert **100** zu.

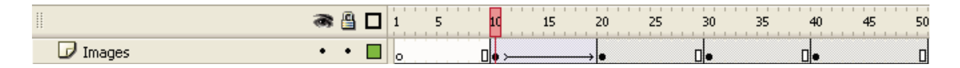
7 Wählen Sie das Schlüsselbild in Bild 30 in der Zeitleiste aus. Wählen Sie dann den „Specials“-Movieclip aus und weisen Sie der *y*-Koordinate im Eigenschafteninspektor den Wert -**200** zu.

8 Wählen Sie das Schlüsselbild in Bild 40 aus. Wählen Sie dann den „Specials“-Movieclip aus und weisen Sie der *y*-Koordinate im Eigenschafteninspektor den Wert -**300** zu.

9 Wählen Sie das Schlüsselbild in Bild 50 aus. Wählen Sie dann den „Specials“-Movieclip aus und weisen Sie der *y*-Koordinate im Eigenschafteninspektor den Wert -**400** zu.

10 Wählen Sie das Schlüsselbild in Bild 10 aus und wählen Sie dann im Eigenschafteninspektor im Popupmenü „Tween“ die Option „Bewegung“ aus.

Dadurch wird die Position des Bilder-Movieclips zwischen den Schlüsselbildern in Bild 10 und Bild 20 getweent.



11 Um fließende Übergänge zwischen den anderen Bildern zu erstellen, wiederholen Sie Schritt 11 für die Schlüsselbilder in Bild 20, 30 und 40.

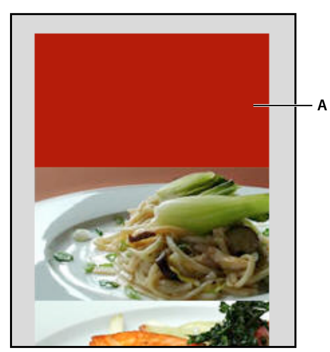
12 Wählen Sie zum Erstellen der Maskenebene in der Zeitleiste die „Specials"-Fotoebene aus und wählen Sie dann „Einfügen" > „Zeitleiste" > „Ebene" aus (oder klicken Sie in der Zeitleiste auf „Ebene einfügen").

13 Doppelklicken Sie auf den Namen der neuen Ebene, und benennen Sie sie in **Maske** um.

14 Wählen Sie Bild 10 der Maskenebene aus, und wählen Sie „Einfügen" > „Zeitleiste" > „Schlüsselbild" aus.

15 Wählen Sie im Bedienfeld „Werkzeuge" das Rechteckwerkzeug aus, und zeichnen Sie über dem ersten (obersten) Bild im Bilder-Movieclip ein Rechteck.

Sie können für das Rechteck eine beliebige Füllfarbe verwenden, aber es muss vollständig undurchsichtig sein.

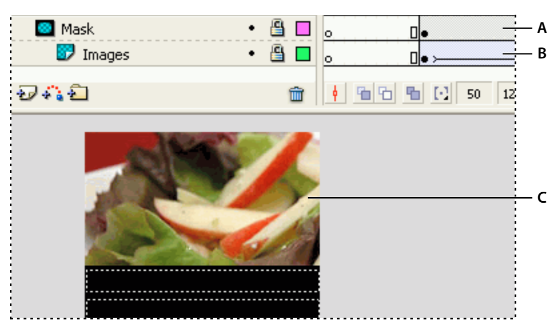


Maskenrechteck

16 Achten Sie darauf, dass das Rechteck den gesamten Bildbereich abdeckt. Um dies zu gewährleisten, klicken Sie zuerst auf das Rechteck, um es auszuwählen. Stellen Sie dann im Eigenschafteninspektor die *x*-Koordinate und die *y*-Koordinate auf **0**, die Breite auf **176** und die Höhe auf **100** ein.

17 Klicken Sie mit der rechten Maustaste (Windows®) bzw. bei gedrückter Taste „Ctrl" (Macintosh®) auf die Bildmaskenebene in der Zeitleiste und wählen Sie im Kontextmenü die Option „Maske" aus.

Die Ebene wird in eine Maskenebene umgewandelt und in der Zeitleiste durch ein Maskenebenensymbol gekennzeichnet. Die unmittelbar darunter liegende Ebene wird mit der Maskenebene verknüpft. Ihr Inhalt ist durch den gefüllten Bereich der Maske hindurch sichtbar. Weitere Informationen zum Arbeiten mit Maskenebenen in Flash finden Sie in den Abschnitten zu Maskenebenen in *Using Flash*.



A. *Maskenebene* ***B.*** *Maskierte Ebene* ***C.*** *Maskierter Bereich auf der Bühne*

18 Speichern Sie die Änderungen („Datei“ > „Speichern“).

Falls Sie die Anwendung jetzt testen, wird die Tagesgericht-Animation nicht abgespielt. Unter „Navigation und Text im Tagesgerichtbildschirm einfügen“ auf Seite 26 können Sie ActionScript hinzufügen, mit dem die Animation bei jedem Schlüsselbild gestoppt wird. Sie können auch Benutzeroberflächenelemente hinzufügen, mit denen der Benutzer die Möglichkeit erhält, zwischen den Bildern zu navigieren.

Navigation und Text im Tagesgerichtbildschirm einfügen

1. Öffnen Sie die Datei cafe_tutorial.fla in Flash.
2. Wählen Sie in der Zeitleiste Bild 10 in der Ebene „Text“ aus.
3. Wählen Sie im Bedienfeld „Werkzeuge“ das Textwerkzeug aus und erstellen Sie unter dem ersten maskierten Tagesgerichtbild ein Textfeld.

   In diesem Textfeld wird der Name des Tagesgerichts angegeben, dessen Bild angezeigt wird.



Textfeld für den Namen des Tagesgerichts

4. Achten Sie darauf, dass das Textfeld auf der Bühne ausgewählt ist, und führen Sie im Eigenschafteninspektor folgende Schritte durch:
   - Wählen Sie im Popupmenü „Texttyp“ die Option „Dynamischer Text“ aus.
   - Geben Sie unter „Instanzname“ **title_txt** ein.
   - Wählen Sie die kursive Schriftstiloption aus.
   - Stellen Sie die Schriftgröße auf 10 ein.
   - Wählen Sie Weiß als Schriftfarbe.
   - Wählen Sie im Popupmenü „Methode zur Schriftwiedergabe“ die Option „Geräteschriftarten verwenden“ aus.
5. Erstellen Sie unterhalb des ersten Textfelds ein zweites, in dem eine kurze Beschreibung des jeweils angezeigten Tagesgerichts dargestellt werden soll.
6. Ändern Sie mithilfe des Auswahlwerkzeugs die Größe des neuen Textfelds so, dass es etwa dreimal so hoch wie das andere Textfeld ist.



Textfeld für die Beschreibung des Tagesgerichts

7 Achten Sie darauf, dass das Textfeld auf der Bühne ausgewählt ist, und führen Sie im Eigenschafteninspektor folgende Schritte durch:

- Wählen Sie im Popupmenü „Texttyp“ die Option „Dynamischer Text“ aus.
- Geben Sie im Textfeld „Instanzname“ den Namen **description_txt** ein.
- Wählen Sie im Popupmenü „Zeilentyp“ die Option „Mehrzeilig“ aus.
- Stellen Sie die Schriftgröße auf 10 ein.
- Achten Sie darauf, dass als Schriftgrad nicht „Kursiv“ ausgewählt ist.
- Wählen Sie Weiß als Schriftfarbe.
- Wählen Sie im Popupmenü „Methode zur Schriftwiedergabe“ die Option „Geräteschriftarten verwenden“ aus.

8 Wählen Sie in der Zeitleiste in der Ebene „ActionScript“ das Schlüsselbild in Bild 10 aus.

9 Öffnen Sie das Bedienfeld „Aktionen“ und geben Sie folgenden Code ein:

```
stop();
fscommand2("SetSoftKeys", "Home", "Next");
title_txt.text = "Summer Salad";
description_txt.text = "Butter lettuce with apples, blood orange segments, gorgonzola, and
raspberry vinaigrette.";
```

Mit diesem Code wird der Name und die Beschreibung des ersten Tagesgerichts in die beiden dynamischen Textfelder eingetragen. Darüber hinaus wird der Abspielkopf im aktuellen Bild angehalten und die Softtasten werden registriert.

10 Wählen Sie in der Ebene „ActionScript“ das Schlüsselbild in Bild 20 aus und geben Sie im Bedienfeld „Aktionen“ folgenden Code ein:

```
stop();
title_txt.text = "Chinese Noodle Salad";
description_txt.text = "Rice noodles with garlic sauce, shitake mushrooms, scallions, and
bok choy.";
```

11 Wählen Sie in der Ebene „ActionScript“ das Schlüsselbild in Bild 30 aus und geben Sie im Bedienfeld „Aktionen“ folgenden Code ein:

```
stop();
title_txt.text = "Seared Salmon";
description_txt.text = "Filet of wild salmon with caramelized onions, new potatoes, and
caper and tomato salsa.";
```

12 Wählen Sie in der Ebene „ActionScript“ das Schlüsselbild in Bild 40 aus und geben Sie im Bedienfeld „Aktionen“ folgenden Code ein:

```
stop();
title_txt.text = "New York Cheesecake";
description_txt.text = "Creamy traditional cheesecake served with chocolate sauce and
strawberries.";
```

13 Wählen Sie in der Ebene „ActionScript“ das Schlüsselbild in Bild 50 aus und geben Sie im Bedienfeld „Aktionen“ folgenden Code ein:

```
gotoAndStop("specials");
```

Dieser Code bewirkt, dass der Abspielkopf zum Anfang der Animationssequenz zurückkehrt. Da das erste und das letzte Bild der Animationssequenz identisch sind, entsteht der Eindruck, dass es sich hier um eine kontinuierliche Animation handelt.

14 Wählen Sie in der Zeitleiste Bild 10 der Ebene „ActionScript“ aus. Öffnen Sie das Bedienfeld „Aktionen“ und geben Sie folgenden Code ein:

```
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        // Handle left soft key event
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        // Handle right soft key event
        play ();
        description_txt.text = "";
        title_txt.text = "";
    }
};
Key.addListener (myListener);
```

Durch Drücken der linken Softtaste wird der Abspielkopf auf den Hauptbildschirm der Anwendung gesetzt. Durch Drücken der rechten Softtaste wechselt die Bildanimation zum nächsten Bild.

Weitere Informationen zur Verwendung von Event-Listenern finden Sie unter „Tastendruck-Ereignisse mithilfe eines Tasten-Listeners verarbeiten“ auf Seite 50.

15 Wählen Sie „Datei“ > „Speichern“.

16 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung in Device Central zu testen.

Wählen Sie die Schaltfläche „Specials“. Die „Specials“-Animation sowie der Text sollten nun korrekt angezeigt werden.

Videobildschirm erstellen

In diesem Abschnitt lernen Sie, wie Sie der Anwendung Video hinzufügen. Außerdem können Sie ActionScript-Code hinzuzufügen, mit dem der Benutzer die Möglichkeit erhält, die Wiedergabe (wiedergeben, anhalten) mithilfe der rechten Softtaste zu steuern.

Flash Lite unterstützt zwei Arten von Video: Gerätevideo und FLV-Dateien (Flash Video). Flash Lite 2.x unterstützt Gerätevideo. Flash Lite 3.x unterstützt Gerätevideo und FLV-Dateien. Um dieses Tutorial abzuschließen, müssen Sie entscheiden, ob Sie Gerätevideo oder eine FLV-Datei verwenden möchten.

Hinweis: *Weitere Informationen zur Videounterstützung finden Sie unter* *„Mit Videos arbeiten“ auf Seite 79.*

Videobildschirm mit Gerätevideo erstellen (Flash Lite 2.x oder Flash Lite 3.x)

1 In Flash, open the cafe_tutorial.fla file you completed in „Tagesgerichtbildschirm erstellen“ auf Seite 23.

2 Open the Library options menu located in the upper-right corner of the Library panel and select New Video.

3 In the Video Properties dialog box, do the following:

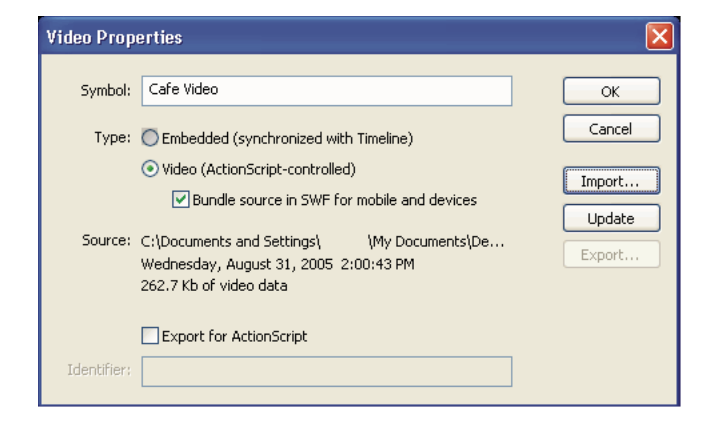
a Enter a name for the video symbol in the Symbol box (**Cafe video**, for example).

b Select Bundle source in SWF for mobile and devices. Click Import.

c Browse to the cafe_townsend_chef.3gp file in the Tutorials folder. Select the file and click Open.

If you don’t see the video file (or if you can see it but can’t select it), select All Files (*.*) from the Files Of Type pop-up menu (Windows), or All Files from the Enable pop-up menu (Macintosh).

4 Make sure the Video Properties dialog box appears as follows, and click OK:



A new video symbol appears in the Library panel.

5 Select frame 51 in the Video layer of the Timeline. Drag the Cafe video symbol from the Library panel to the Stage.

6 In the Property inspector, enter **cafeVideo** in the Instance Name box. Set the *x* position to **0**, the *y* position to **30**, the Width to **176**, and the Height to **144**.



7 In the Timeline, select Frame 51 of the ActionScript layer. Enter, or copy and paste, the following code in to the Actions panel:

```
// Stop timeline, register soft keys, and start video.
stop ();
fscommand2 ("SetSoftKeys", "Home", "Pause");
cafeVideo.play ();
var playing:Boolean = true;
// Soft key event handler code:
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        if (playing) {
            // If video is playing, pause it,
            // set status variable (playing) to false,
            // and set right soft key label to 'Play'.
            cafeVideo.pause ();
            playing = false;
            rightSoftKeyLabel.text = "Play";
        }
        else {
            // If video is paused, resume its playback,
            // set status variable (playing) to true,
```

```
            // and set right soft key label to 'Pause'.
            cafeVideo.resume ();
            playing = true;
            righttSoftKeyLabel.text = "Pause";
        }
    }
};
// Register listener object:
Key.addListener (myListener);
// Video status handler code.
// Device Central does not support the onStatus handler.
cafeVideo.onStatus = function (infoObject:Object) {
    var code = infoObject.code;
    if (code == "completed") {
        // If video has finished playing, set playing=false,
        // and set right soft key label to "Replay":
        playing = false;
        rightSoftKeyLabel.text = "Replay";
    }
};
```

8 Choose File > Save.

9 Choose Test > Control Movie to test the application in Device Central.

Select View Video on the homescreen to view the video. Press the Right soft key to pause the video. Press the Right soft key again to resume playback. On a device, you can press the Right soft key to play the video again after playback completes. However, this feature (the `Video.onStatus` handler) is not supported in Device Central.

Videobildschirm mit einer FLV-Datei erstellen (Flash Lite 3.x)

1 In Flash, open the cafe_tutorial.fla file you completed in „Tagesgerichtbildschirm erstellen“ auf Seite 23.

2 Locate the cafe_townsend_chef.flv file in the Tutorials folder. Copy this file to the same folder as the cafe_tutorial.fla file.

3 Open the Library options menu located in the upper-right corner of the Library panel and select New Video.

4 In the Video Properties dialog box, do the following:

a Enter a name for the video symbol in the Symbol box (**Cafe video**, for example).

b Select Video (ActionScript-Controlled). Click OK.

5 Select Frame 51 in the Video layer of the Timeline. Drag the Cafe video symbol from the Library panel to the Stage.

6 In the Property inspector, enter **cafeVideo** in the Instance Name box. Set the *x* position to **0**, the *y* position to **45**, the Width to **176**, and the Height to **144**.



7 In the Timeline, select Frame 51 of the ActionScript layer. Enter, or copy and paste, the following code in to the Actions panel:

```
// Frame actions
stop();
fscommand2 ("SetSoftKeys", "Home", "Pause");
var videoIsPlaying:Boolean;
// Use the NetConnection and NetStream classes to play FLV files
var nc:NetConnection = new NetConnection();
var ns:NetStream;
/* When a change in connection status occurs,
Flash Lite triggers the onStatus handler and passes it an
object with information about the change.
Use the information object to run code when
connection events occur.
*/
nc.onStatus = function(info) {
    // Run the following code if the connection is successful.
    if(info.code == "NetConnection.Connect.Success"){
        // Create a NetStream object and pass it the NetConnection object
        ns = new NetStream(nc);
        // Attach the NetStream object to the video object on Stage.
        cafeVideo.attachVideo(ns);
        // Play the file "cafe_townsend_chef.flv".
        ns.play("cafe_townsend_chef.flv");
        videoIsPlaying = true;
    }
}
// Create a connection.
// To connect to a local file or to a file on a web server, pass null.
nc.connect(null);
/* When a change in stream status occurs,
Flash Lite triggers the onStatus handler and passes it an
object with information about the change.
Use the information object to run code when
stream changes occur.
*/
ns.onStatus = function(info){
    if(info.code == "NetStream.Play.Stop"){
        videoIsPlaying = false;
        rightSoftKeyLabel.text = "Play";
    }
};
// Capture and handle soft key events.
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    // Run this code if the user presses the left soft key.
    if (keyCode == ExtendedKey.SOFT1) {
        // Close the video.
        ns.close();
        // Go to the home screen.
        gotoAndPlay ("home");
    }
    // Run this code if the user presses the right soft key
    else if (keyCode == ExtendedKey.SOFT2) {
```

```
        // If the video is playing, run this code
        if (videoIsPlaying){
            // Pause or restart the video (this method does both).
            ns.pause();
            // Switch the label text.
            if (rightSoftKeyLabel.text == "Play"){
                rightSoftKeyLabel.text = "Pause";
            }
            else if (rightSoftKeyLabel.text == "Pause"){
                rightSoftKeyLabel.text = "Play";
            }
        // If the video has finished playing, run this code.
        } else {
            // Replay the video.
            ns.play("cafe_townsend_chef.flv");
            rightSoftKeyLabel.text = "Pause";
            videoIsPlaying = true;
        }
    }
};
Key.addListener (myListener);
```

8 Choose File > Save.

9 Choose Test > Control Movie to test the application in Device Central.

Select View Video on the homescreen to view the video. Press the Right soft key to pause the video. Press the Right soft key again to resume playback. When the video has finished playing, press the Right soft key to play it again.

Standortauswahlbildschirm erstellen.

In diesem Abschnitt erstellen Sie einen neuen Bildschirm, in dem Benutzer auswählen können, welcher der vier Standorte des Restaurants für eine Reservierung angerufen werden soll.

Hinweis: *Wählen Sie zum Anzeigen der gewählten Nummer „Fenster“ > „Flash-Ausgabe“ in Device Central CS4 oder „Ansicht“ > „Flash-Ausgabe“ > „Anzeigen“ in Device Central CS3.*

Wenn der Benutzer die Anwendung das erste Mal startet und „Reservations“ auswählt, zeigt die Anwendung den Standortauswahlbildschirm an. Wenn der Benutzer also „Reservations“ auswählt, wählt die Anwendung den ausgewählten Standort an. Die Anwendung verwendet ein gemeinsames Objekt, um den Standort zwischen Sitzungen permanent zu speichern.

1 Öffnen Sie in Flash die Datei cafe_tutorial.fla, die Sie im Abschnitt „Videobildschirm erstellen“ auf Seite 29 fertig gestellt haben.

2 Wählen Sie in der Zeitleiste in der Ebene "Options Menu" das Schlüsselbild in Bild 66 aus.

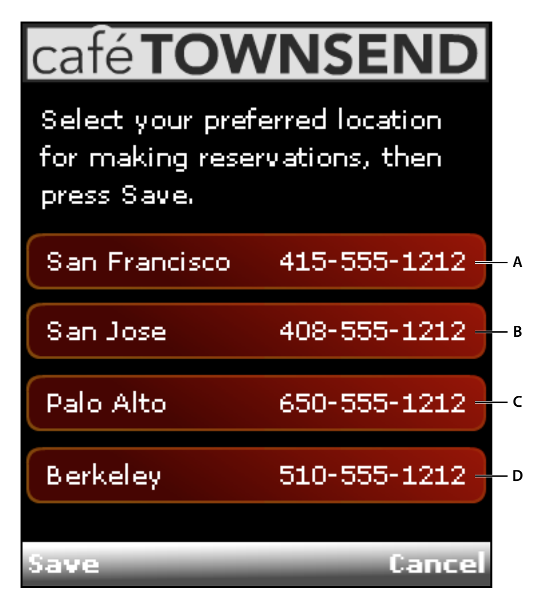
3 Ziehen Sie die Schaltfläche „location_SF“ von der Bibliothek auf die Bühne. Geben Sie im Eigenschafteninspektor im Feld „Instanzname “ den Namen **sf_btn** ein.

4 Ziehen Sie die Schaltfläche „location_SJ“ von der Bibliothek auf die Bühne. Geben Sie im Eigenschafteninspektor im Feld „Instanzname “ den Namen **sj_btn** ein.

5 Ziehen Sie die Schaltfläche „location_PA“ von der Bibliothek auf die Bühne. Geben Sie im Eigenschafteninspektor im Feld „Instanzname “ den Namen **pa_btn** ein.

6 Ziehen Sie die Schaltfläche „location_BK“ von der Bibliothek auf die Bühne. Geben Sie im Eigenschafteninspektor im Feld „Instanzname “ den Namen **bk_btn** ein.

Die Bühne der Anwendung sollte ungefähr folgendermaßen aussehen:



***A.** sf_btn **B.** sj_btn **C.** pa_btn **D.** bk_btn*

7 Wählen Sie Bild 61 in der Ebene „ActionScript“ aus. Geben Sie im Bedienfeld „Aktionen“ folgenden Code ein:

```
/* Frame Actions */
stop ();
fscommand2 ("SetSoftKeys", "Save", "Cancel");
setLocation ();

/* Handle soft key presses */
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1 ) {
        // Save location
        saveNewLocation ();
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        // Cancel operation, go back to home screen
        gotoAndPlay ("home");
    }
};
Key.addListener (myListener);

/*
The savenewLocation() function is called when the user presses "Save" (left soft key).
It updates the "location" shared object with the phone # associated with that location.
*/
function saveNewLocation () {
    // Determine which button (location) the user selected:
    var selectedButton = Selection.getFocus ();
    switch (selectedButton) {
        case "_level0.sf_btn" :
            // User selected San Francisco
            location_so.data.phoneNumber = "415-555-1212";
```

```
        break;
        case "_level0.sj_btn" :
            // User selected San Jose
            location_so.data.phoneNumber = "408-555-1212";
        break;
        case "_level0.bk_btn" :
            // User selected Berkeley
            location_so.data.phoneNumber = "510-555-1212";
        break;
        case "_level0.pa_btn" :
            // User selected Palo Alto
            location_so.data.phoneNumber = "650-555-1212";
        break;
    }
}

/*
The setLocation() function sets the section to
the currently selected location,
based on the phone number saved in the shared object.
*/
function setLocation () {
    // Retreive phone number stored in shared object:
    var loc:String = location_so.data.phoneNumber;
    // Extract area code from phone number:
    var areaCode:String = loc.substring (0, 3);
    // Based on area code, set selection focus to corresponding button/menu item:
    switch (areaCode) {
        case "415" :
            Selection.setFocus (_level0.sf_btn);
        break;
        case "408" :
            Selection.setFocus (_level0.sj_btn);
        break;
        case "510" :
            Selection.setFocus (_level0.bk_btn);
        break;
        case "650" :
            Selection.setFocus (_level0.pa_btn);
        break;
    }
}
```

8 Wählen Sie „Datei“ > „Speichern“.

9 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung in Device Central zu testen.

Öffnen Sie das Fenster „Ausgabe“ in Device Central zur Anzeige des angewählten Standorts.

Kapitel 4: Interaktivität und Navigation erstellen

Zur Interaktion mit Ihrer Adobe Flash Lite-Anwendung muss der Benutzer bestimmen können, welches Objekt auf dem Bildschirm derzeit den Fokus hat. Außerdem muss er in der Lage sein, zwischen Objekten zu navigieren und durch Auswahl eines Objekts oder einer anderen Taste eine Aktion auszulösen. Dieselben Grundsätze gelten zwar auch für Desktop-Anwendungen, doch variieren manche Funktionen bei Mobilgeräten.

Benutzerinteraktion und unterstützte Tasten

Benutzerinteraktion in Flash Lite

Flash Lite unterstützt die Navigation und Benutzerinteraktion, die je nach Geräteausstattung über die Tastatur des Geräts, über einen Stift oder über einen Touchscreen erfolgen. Die für Ihre Anwendung verfügbaren Optionen richten sich nach dem Zielgerät und dem Inhaltstyp. Weitere Informationen über Inhaltstypen finden Sie unter „Flash Lite-Inhaltstypen“ auf Seite 17.

Am einfachsten können Sie eine Flash Lite-Anwendung über die Standardnavigation mit tastenbasierter Interaktivität versehen. Dabei wird die Vierwege-Tastatur des Geräts verwendet, z. B. die Pfeiltasten, die Tabulatortaste und die Kombination aus Umschalt- und Tabulatortaste in einer Desktop-Anwendung. Der Benutzer bewegt den Fokus zum gewünschten Objekt und drückt dann die Auswahltaste. Die Anwendung enthält Ereignisprozedur-Code zur Reaktion auf diese Schaltflächenereignisse. Die Standardnavigation funktioniert in Flash Lite bei Schaltflächen, Eingabetextfeldern und wahlweise auch bei Movieclips. Sie sollte nach Möglichkeit für einfache Benutzerinteraktionen wie Menüs verwendet werden. Weitere Informationen über Standardnavigation finden Sie unter „Standardnavigation in Flash Lite verwenden“ auf Seite 38.

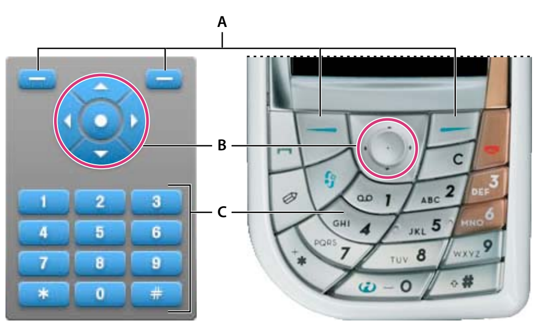
Anwendungen können auch auf beliebige Tastendruck-Ereignisse reagieren, die von Flash Lite generiert werden, wenn der Benutzer eine bestimmte Taste drückt. Mithilfe der ereignisbasierten Navigation können Sie Flash Lite-Anwendungen mit einem komplexen Benutzerinteraktionsmodell erstellen, beispielsweise Spiele. Weitere Informationen über Ereignisse finden Sie unter „Tasten- und Schaltflächenereignisse“ auf Seite 44.

Von Flash Lite unterstützte Tasten

Abgesehen von den alphanumerischen Tasten, mit denen herkömmliche Telefone ausgestattet sind, haben die meisten Mobilgeräte eine Navigationstastatur, die zur Navigation und zur Auswahl von Elementen auf dem Gerätebildschirm dient, sowie mindestens zwei Softkeys. Die *Softkeys* eines Geräts sind multifunktionale Tasten, deren aktuelle Funktion jeweils auf dem Gerätebildschirm angegeben wird.

Eine typische Navigationstastatur besteht aus vier Navigationstasten (Nach oben, Nach unten, Nach links und Nach rechts) und einer Auswahltaste (befindet sich in der Regel in der Mitte der Tastatur). Wie diese Tasten eingesetzt werden, hängt von der jeweiligen Anwendung ab. Beispielsweise kann in einem Spiel eine Spielfigur auf dem Bildschirm mit den Navigationstasten in Bewegung gesetzt und mit der Auswahltaste zum Springen gebracht werden.

Die folgenden Abbildungen zeigen die häufigsten Tasten einer generischen Tastatur und eines bestimmten Geräts:



*A. Linker und rechter Softkey B. Navigationstastatur C. Zifferntasten, * und #*

Nicht alle Geräte und Flash Lite-Inhaltstypen unterstützen sämtliche Tasten. Beispielsweise bieten Geräte, welche die Zweiwege-Navigation unterstützen, keine Unterstützung für die linke und rechte Navigationstaste (siehe„Standardnavigationsmodi“ auf Seite 38). Auch stehen die Softkeys des Geräts nicht immer zur Verfügung.

Flash Lite unterstützt die folgenden Tasten auf Mobilgeräten:

<table>
<tr><th>Beschreibung</th><th>Tasten</th><th>Verfügbarkeit</th></tr>
<tr><td>Zifferntasten, *, #</td><td>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #</td><td>Alle Geräte</td></tr>
<tr><td>Fünfwege-Tastatur</td><td>Auswahl, Nach oben, Nach unten</td><td>Alle Geräte</td></tr>
<tr><td></td><td>Nach links, Nach rechts</td><td>Nur Geräte, die Vierwege-Navigation unterstützen. (siehe „Standardnavigationsmodi“ auf Seite 38)</td></tr>
<tr><td>Softkeys</td><td>Nach links, Nach rechts</td><td>Geräte, die den Befehl <code>SetSoftKeys</code> unterstützen</td></tr>
<tr><td></td><td>Tasten SOFT3 - SOFT12</td><td>Geräte, die mehr als zwei Softkeys haben</td></tr>
<tr><td>Tastaturtasten</td><td>!, “, #, $, %, &, ‘, (,), *, +, ,, -, ., /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, +, >, ?, @, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, [, \,], ^, _, ‘, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, {, |, }, ~, Rücktaste</td><td>Geräte mit einer QWERTY-Tastatur</td></tr>
</table>

Mit den folgenden System.Capabilities-Eigenschaften können Sie ermitteln, welche Navigations- und Auswahloptionen bei einem bestimmten Gerät zur Verfügung stehen:

- `hasMappableSoftKeys`
- `softKeyCount`
- `has4WayKeyAS`
- `hasQWERTYKeyboard`
- `hasStylus`
- `hasMouse`

Weitere Informationen zur System.Capabilites-Klasse finden Sie im *Flash Lite 2.x und 3 ActionScript-Referenzhandbuch*.

Standardnavigation

Standardnavigation in Flash Lite verwenden

In Flash-Desktop-Anwendungen können Benutzer mit der Tabulatortaste und der Tastenkombination <Umschalt > + <Tab > den Fokus von einem Objekt auf dem Bildschirm auf ein anderes verlagern. Die Pfeiltasten funktionieren in einigen anderen Anwendungen auf ähnliche Weise. In Flash Lite erfüllen die Navigationstasten auf der Navigationstastatur eines Geräts denselben Zweck wie die Pfeiltasten oder die Tastenkombination <Umschalt > + <Tab > in einer Flash-Desktop-Anwendung. Sobald das gewünschte Objekt den Fokus hat, kann der Benutzer die Auswahltaste drücken, um eine Aktion in der Anwendung auszulösen. Als Reaktion auf die Auswahl einer Schaltfläche oder eines Movieclips schreiben Sie Ereignisprozeduren. Weitere Informationen hierzu finden Sie unter „Schaltflächenereignisse verarbeiten“ auf Seite 44.

In Flash Lite kann die Standardnavigation bei Schaltflächen und Textfeldern verwendet werden. Dies ist auch bei Movieclips der Fall, deren Eigenschaft `tabEnabled` auf `true` eingestellt ist oder denen Ereignisprozeduren zugewiesen sind, wobei die Eigenschaft `tabEnabled` nicht auf `false` eingestellt sein darf.

Wenn ein Eingabetextfeld den Fokus hat und der Benutzer die Auswahltaste drückt, öffnet Flash Lite das generische Texteingabe-Dialogfeld des Geräts, in dem der Benutzer Text eingeben kann.

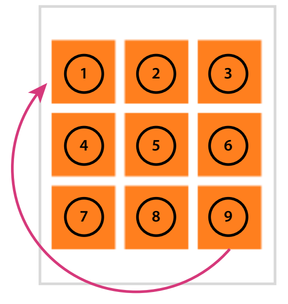
Ein Beispiel für Standardnavigation finden Sie unter „Einfaches Menü mithilfe von Schaltflächen und Standardnavigation erstellen“ auf Seite 47.

Standardnavigationsmodi

Flash Lite unterstützt drei Standardnavigationsmodi: Zweiwege, Vierwege und Vierwege mit Rücksprungfunktion. Welche Navigationsmodi unterstützt werden, hängt jeweils vom Gerät und vom Flash Lite-Inhaltstyp ab. Wie Sie die Navigationsmodi für bestimmte Geräte und Inhaltstypen ermitteln, erfahren Sie unter „Gerätevideo im Emulator wiedergeben“ auf Seite 125.

Zweiwege-Navigation in Flash Lite entspricht der Tabulatornavigation in Flash-Desktop-Anwendungen, in denen der Benutzer mit der Tabulatortaste und der Tastenkombination <Umschalt > + <Tab > zwischen Objekten auf dem Bildschirm navigieren kann. Die Nach-unten-Navigationstaste auf dem Gerät entspricht der Tabulatortaste und die Nach-oben-Navigationstaste der Kombination aus Umschalt- und Tabulatortaste.

Die Tabulatorreihenfolge in der Zweiwege-Navigation ist standardmäßig von links nach rechts und von oben nach unten. Die folgende Abbildung zeigt beispielsweise ein aus je drei Reihen und Spalten bestehendes Raster mit Schaltflächenobjekten in einer Flash Lite-Anwendung. Die Zahlen über den Schaltflächen geben die Reihenfolge an, in der die einzelnen Schaltflächen den Tastaturfokus erhalten, wenn der Benutzer wiederholt die Navigationstaste <Nach-unten> des Geräts drückt. Nachdem die Schaltfläche unten rechts den Tastaturfokus erhalten hat, springt der Fokus wieder zur Schaltfläche oben links zurück, wenn der Benutzer das nächste Mal die Navigationstaste <Nach-unten > drückt.



Beispiel für die Tabulatorreihenfolge bei der Zweiwege-Navigation

Sie können die Tabulatorreihenfolge für die Zweiwege-Navigation mithilfe der Eigenschaft `tabIndex` von Button-, MovieClip- und TextField-Objekten anpassen. Weitere Informationen finden Sie unter „Steuern der Tabulatorreihenfolge bei der Zweiwege-Navigation“ auf Seite 43.

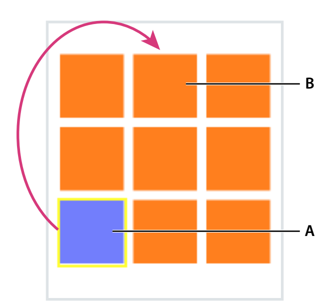
Ein Beispiel für eine Zweiwege-Navigation finden Sie auf der Flash Lite-Website auf der Seite mit Beispielen und Tutorials unter www.adobe.com/go/learn_flt_samples_and_tutorials_de.Laden Sie die komprimierte Datei für Ihre ActionScript-Version herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf die Beispieldatei 2-way.fla zuzugreifen.

Vierwege-Navigation in Flash Lite ähnelt der Verwendung der Pfeiltasten auf einer PC-Tastatur zur Navigation zwischen den Objekten auf dem Bildschirm. Die Navigationstasten <Nach-oben >, <Nach-unten >, <Nach-links > und <Nach-rechts > des Geräts entsprechen den vier Pfeiltasten einer Computertastatur. Durch Drücken einer Navigationstaste wird der Tastaturfokus auf das Objekt verlagert, das sich in dieser Richtung befindet. Wenn in dieser Richtung kein Objekt vorhanden ist, bleibt der Tastaturfokus auf dem aktuellen Objekt.

Hinweis: *Geräte mit Unterstützung für die Vierwege-Navigation unterstützen nicht die Eigenschaft* `tabIndex`*, aber die Eigenschaften* `tabEnabled` *und* `tabChildren`*. Dies unterscheidet sich von der Funktionsweise dieser Eigenschaften in Flash-Desktop-Anwendungen.*

Ein Beispiel für eine Vierwege-Navigation finden Sie auf der Flash Lite-Website auf der Seite mit Beispielen und Tutorials unter www.adobe.com/go/learn_flt_samples_and_tutorials_de. Laden Sie die komprimierte Datei für Ihre ActionScript-Version herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf die Beispieldatei „4-way.fla“ zuzugreifen.

Vierwege-Navigation mit Rücksprungfunktion entspricht einer Kombination aus der standardmäßigen Vierwege- und Zweiwege-Navigation. Wie bei der bereits beschriebenen standardmäßigen Vierwege-Navigation kann der Benutzer den Tastaturfokus über die Vierwege-Navigationstasten des Geräts ändern. Der Unterschied ist jedoch, dass der Tastaturfokus zum Objekt auf der gegenüberliegenden Seite des Bildschirms zurückspringt, ähnlich wie bei der Zweiwege-Navigation. Im unten gezeigten Beispiel hat die Schaltfläche unten links den Tastaturfokus. Wenn der Benutzer nun die Nach-unten-Navigationstaste drückt, erhält die Schaltfläche in der Mitte der oberen Schaltflächenreihe den Fokus.



A. *Schaltfläche mit Fokus* ***B.*** *Nächste Schaltfläche, die den Fokus erhält, wenn der Benutzer die Navigationstaste drückt*

Sie können das Verhalten der Zwei- und der Vierwege-Navigation im Adobe Device Central-Emulator mit den Beispieldateien www.adobe.com/go/learn_flt_samples_and_tutorials_de testen. Suchen Sie auf der Seite mit Beispielen und Tutorials nach der komprimierten Datei für Ihre Flash Lite-Version, laden Sie sie herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf das Beispiel zuzugreifen. Beide Beispieldateien enthalten das bereits beschriebene Schaltflächengitter mit je drei Reihen und Spalten. Der einzige Unterschied zwischen den Beispieldateien besteht darin, dass jede FLA-Datei für eine Kombination aus Gerät und Flash Lite-Inhaltstyp konfiguriert ist, die den jeweiligen Navigationsmodus unterstützt (Zweiwege oder Vierwege).

Wenn Sie sich eine Beispieldatei näher ansehen möchten, öffnen Sie sie in Flash und testen Sie sie im Adobe Device Central-Emulator (wählen Sie Steuerung > Film testen). Klicken Sie auf die Pfeiltasten, die auf der Tastatur des Emulators abgebildet sind (oder drücken Sie die Pfeiltasten auf der Computertastatur), um die Wirkung der einzelnen Navigationsmodi zu verfolgen.

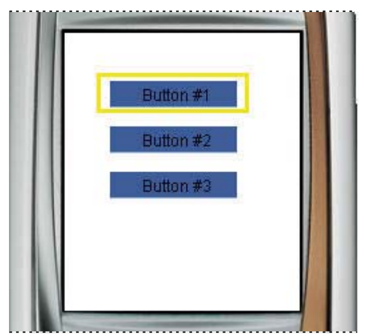


Schaltflächen in einer Beispieldatei

Das Fokusrechteck

In Flash Lite werden Schaltflächen oder Eingabetextfelder, die zurzeit den Fokus haben, standardmäßig mit einem gelben Rechteck gekennzeichnet. Dies ist auch bei Movieclips der Fall, deren Eigenschaft `tabEnabled` auf `true` eingestellt ist oder denen Ereignisprozeduren zugewiesen sind, wobei die Eigenschaft `tabEnabled` nicht auf `false` eingestellt sein darf.

Anhand des Fokusrechtecks kann der Benutzer erkennen, welches Objekt auf dem Bildschirm reagiert, wenn er die Auswahltaste des Geräts drückt. In der folgenden Abbildung umgibt das Fokusrechteck z. B. eine Schaltfläche, die gegenwärtig den Tastaturfokus hat:



Schaltfläche mit aktuellem Fokus und standardmäßigem Fokusrechteck

Bei Schaltflächen und Movieclips hängt die Größe des Begrenzungsrahmens des Fokusrechtecks vom *Kollisionsbereich* des Objekts ab. Dies ist der unsichtbare Bereich, der (in Flash-Desktop-Anwendungen) den Teil der Schaltfläche oder des Movieclips definiert, der auf Mausklicks reagiert. Bei Eingabetextfeldern hingegen ergibt sich die Größe des Begrenzungsrahmens des Fokusrechtecks aus den Abmessungen des Textfelds.

Sie können das Fokusrechteck deaktivieren oder seine Farbe ändern. Weitere Informationen finden Sie unter „Das Fokusrechteck anpassen“ auf Seite 41.

Richtlinien für die Standardnavigation

Im Folgenden finden Sie einige Richtlinien und Aspekte, die bei der Verwendung der Standardnavigation in Flash Lite-Anwendungen berücksichtigt werden sollten.

- Wenn Sie das standardmäßige Fokusrechteck deaktivieren, indem Sie `_focusRect` auf `false` einstellen, müssen Sie einen anderen Fokusindikator für Schaltflächen, Eingabetextfelder und Movieclips mit aktivierter Eigenschaft „tabEnabled" verwenden. Bei Schaltflächen können Sie zu diesem Zweck ein visuelles Element an den Over-Status der Schaltfläche anfügen. Dies ist der Teil der Zeitleiste des Schaltflächenobjekts, der angezeigt wird, wenn die Schaltfläche den Fokus hat. Ein Beispiel für diese Technik finden Sie in der Beispielanwendung unter „Einfaches Menü mithilfe von Schaltflächen und Standardnavigation erstellen" auf Seite 47. Bei Eingabetextfeldern können Sie mit dem Selection-Objekt bestimmen, wann das Textfeld den Fokus erhält und den gewünschten Fokusindikator einblenden. Ein Beispiel finden Sie in der Beispielanwendung, die unter „Fokus mithilfe von ActionScript steuern" auf Seite 42 beschrieben wird.
- Auf dem Bildschirm müssen sich stets mindestens zwei Objekte befinden (Eingabetextfelder, Schaltflächen, Movieclips mit aktivierter Eigenschaft tabEnabled oder eine Kombination). Wenn der Bildschirm nur eines dieser Objekte enthält, kann der Benutzer den Fokus nicht ändern und dies kann den Eindruck erwecken, dass er in der Benutzeroberfläche <:glq>festsitzt <:grq>.

  Wenn ein Bildschirm in einer Anwendung nur eine einzige Schaltfläche zur Benutzerinteraktion enthält, bietet es sich an, anstelle von Schaltflächenereignissen ein Tastendruck-Ereignis zu verwenden. Weitere Informationen finden Sie unter „Tasten- und Schaltflächenereignisse" auf Seite 44.
- In manchen Fällen empfiehlt es sich, die Methode `Selection.setFocus()` zu verwenden, um den ursprünglichen Fokus auf ein bestimmtes Objekt auf dem Bildschirm zu legen. Dies kann dem Benutzer die Navigation in der Oberfläche erleichtern und den Navigationsaufwand über Tasten reduzieren. Angenommen, ein Bildschirm Ihrer Anwendung enthält ein Texteingabefeld. Zur Eingabe eines Wertes in das Textfeld muss der Benutzer normalerweise erst eine Navigationstaste drücken, damit das Textfeld den Fokus erhält und dann die Auswahltaste, um das Dialogfeld für die Texteingabe zu öffnen. Mit dem folgenden ActionScript-Code können Sie festlegen, dass das Eingabetextfeld automatisch den Tastaturfokus erhält:

  ```
  Selection.setFocus(inputTxt);
  ```

  Weitere Informationen zur Fokussteuerung mithilfe von ActionScript finden Sie unter „Fokus mithilfe von ActionScript steuern" auf Seite 42.
- Die Ausrichtung der Objekte auf dem Bildschirm spielt bei der Standardnavigation eine wichtige Rolle. Wenn die Objekte auf dem Bildschirm nicht nebeneinander angeordnet sind, ist die Reihenfolge, in der sie den Fokus erhalten, für Benutzer wahrscheinlich nicht offensichtlich. Sie können dies verhindern, indem Sie Objekte nach Möglichkeit mit einheitlicher Größe erstellen und vertikal und horizontal ausrichten. Bei der Zweiwege-Navigation können Sie die Reihenfolge auch mit der Eigenschaft `tabIndex` steuern. Weitere Informationen hierzu finden Sie unter „Steuern der Tabulatorreihenfolge bei der Zweiwege-Navigation" auf Seite 43.

Das Fokusrechteck anpassen

Das Fokusrechteck ist standardmäßig ein gelber Rahmen, der die ausgewählte Schaltfläche oder das ausgewählte Eingabetextfeld umgibt. Dies ist auch bei Movieclips der Fall, deren Eigenschaft `tabEnabled` auf `true` eingestellt ist oder denen Ereignisprozeduren zugewiesen sind, wobei die Eigenschaft `tabEnabled` nicht auf `false` eingestellt sein darf. Weitere Informationen finden Sie unter „Das Fokusrechteck" auf Seite 40.

Sie können das standardmäßige Fokusrechtseck deaktivieren, indem Sie die globale Eigenschaft `_focusRect` auf `false` einstellen. Außerdem haben Sie die Möglichkeit, das Fokusrechteck für bestimmte Schaltflächen oder Movieclips zu deaktivieren (siehe `focusrect (Button._focusrect-Eigenschaft)` und `focusrect (MovieClip.focusrect-Eigenschaft)`) im*Flash Lite 2.x und 3 ActionScript-Referenzhandbuch.*

Auch die Standardfarbe Gelb des Fokusrechtecks kann in eine beliebige andere Farbe geändert werden. Dazu verwenden Sie den Befehl `SetFocusRectColor` mit RGB-Werten als Parameter. Mit dem folgenden Code wird die Farbe des Fokusrechtecks beispielsweise in Rot geändert:

```
fscommand2("SetFocusRectColor", 255, 0, 0);
```

Fokus mithilfe von ActionScript steuern

Mit dem Selection-Objekt von ActionScript können Sie den aktuellen Tastaturfokus abrufen und einstellen. Außerdem können Sie mit diesem Objekt festlegen, dass Sie eine Benachrichtigung erhalten, wenn ein Objekt den Tastaturfokus erhält oder verliert. Dies ist beispielsweise nützlich, wenn der Fokus beim Laden der Anwendung automatisch auf eine bestimmte Schaltfläche gesetzt werden soll. Möglicherweise möchten Sie auch benachrichtigt werden, wenn ein bestimmtes Objekt auf dem Bildschirm den Tastaturfokus erhält oder verliert, damit Sie den Bildschirm entsprechend aktualisieren können.

Im folgenden Beispielcode wird die Methode `Selection.setFocus()` verwendet, um den Fokus auf die Schaltflächeninstanz `login_btn` zu setzen:

```
Selection.setFocus(login_btn);
```

Mit dem Ereignis-Listener `Selection.onSetFocus` können Sie bestimmen, wann der Tastaturfokus sich geändert hat. Sie können diesen Ereignis-Listener beispielsweise verwenden, um einen benutzerdefinierten Fokusmanager für Eingabetextfelder zu erstellen, der anstelle des standardmäßigen Fokusrechtecks verwendet werden soll. Im Folgenden wird gezeigt, wie Sie einen benutzerdefinierten Fokusmanager erstellen, der die Rahmenfarbe des TextField-Objekts ändert, das den Fokus hat. Ein Beispiel der vollständigen Anwendung (custom_focus_manager.fla) finden Sie auf der Flash Lite-Website auf der Seite mit Beispielen und Tutorials unter www.adobe.com/go/learn_flt_samples_and_tutorials_de.Laden Sie die komprimierte Datei für Ihre ActionScript-Version herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf das Beispiel zuzugreifen.

Einen benutzerdefinierten Fokusmanager für Texteingabefelder erstellen

1 Erstellen Sie in Flash ein neues Mobilgerätdokument, und speichern Sie es unter custom_focus_manager.fla

2 Erstellen Sie mit dem Textwerkzeug ein Textfeld auf der Bühne.

3 Achten Sie darauf, dass das Textfeld ausgewählt ist und führen Sie dann im Eigenschafteninspektor die folgenden Schritte aus: Wählen Sie im Popupmenü „Texttyp“ die Option „Eingabetext“ aus, geben Sie **inputTxt_1** in das Textfeld „Instanzname“ ein und aktivieren Sie die Option „Rahmen um Text zeigen“.

4 Erstellen Sie auf dieselbe Weise unter dem ersten Textfeld ein weiteres Textfeld mit dem Instanznamen **inputTxt_2** und aktivieren Sie auch für dieses Textfeld die Option „Rahmen um Text zeigen“.

5 Wählen Sie in der Zeitleiste Bild 1 der Ebene ActionScript aus.

6 Öffnen Sie das Bedienfeld Aktionen (Fenster > Aktionen) und geben Sie folgenden Code ein (Sie können den Code auch kopieren und einfügen):

```
// Disable focus rect globally:
_focusrect = false;
// Create Selection listener object:
var focusListener:Object = new Object ();
// Define onSetFocus method:
focusListener.onSetFocus = function (oldFocus, newFocus) {
    // Enable/disable selection indicator:
    if (newFocus instanceof TextField) {
        // Set border color of text field with new focus to red:
        newFocus.borderColor = 0xFF0000;
    }
    if (oldFocus != undefined && oldFocus instanceof TextField) {
        // Set border color of text field with old focus to black:
        oldFocus.borderColor = 0x000000;
    }
};
// Add listener to Selection object:
Selection.addListener (focusListener);
// Set initial focus when application loads:
Selection.setFocus (inputTxt_1);
// Enable full-screen mode:
fscommand2 ("FullScreen", true);
```

7 Speichern Sie Ihre Änderungen und testen Sie die Anwendung im Emulator (Steuerung > Film testen).

8 Drücken Sie die Nach-oben- und Nach-unten-Taste des Emulators, um den Tastaturfokus zwischen den beiden Textfeldern zu verschieben. Das Textfeld mit Fokus sollte einen roten Rahmen haben, das Textfeld ohne Fokus dagegen einen schwarzen Rahmen. Drücken Sie die Auswahltaste, wenn ein Textfeld den Fokus hat, um das Dialogfeld für die Texteingabe einzublenden.

Steuern der Tabulatorreihenfolge bei der Zweiwege-Navigation

Die Zweiwege-Navigation in Flash Lite entspricht der Tabulatornavigation in Flash und unterstützt deshalb die Eigenschaft `tabIndex`, mit der Sie die Tabulatorreihenfolge von Schaltflächen, Movieclips und Eingabetextfeldern explizit festlegen können. Bei Geräten mit Unterstützung für die Vierwege-Navigation wird die Eigenschaft `tabIndex` nicht unterstützt. Daher kann die Tabulatorreihenfolge für die Vierwege-Navigation nicht mit der Eigenschaft `tabIndex` eingestellt werden.

Zur Steuerung der Tabulatorreihenfolge in der Zweiwege-Navigation weisen Sie der Eigenschaft `tabIndex` eines jeden Objekts eine Zahl zu, die die Reihenfolge des jeweiligen Objekts in der Standardnavigation angibt. Angenommen, eine Anwendung enthält eine Schaltfläche (`my_button`), einen Movieclip (`my_movieclip`) und ein Eingabetextfeld (`my_inputTxt`). Mit dem folgenden Code wird die Tabulatorreihenfolge so festgelegt, dass zunächst die Schaltfläche, dann der Movieclip und schließlich das Eingabetextfeld den Fokus erhält.

```
my_button.tabIndex = 1;
my_movieclip.tabEnabled = true;
my_movieclip.tabIndex = 2;
my_inputTxt.tabIndex = 3;
```

Tasten- und Schaltflächenereignisse

Tasten- und Schaltflächenereignisse

Ereignisprozeduren und Ereignis-Listener bestimmen, wie die Anwendung auf Ereignisse reagiert, die vom Benutzer oder vom System ausgelöst werden. Wenn beispielsweise eine Schaltfläche den Fokus hat und der Benutzer die Auswahltaste drückt, wird das Ereignis `onPress` generiert. Flash Lite-Anwendungen können nicht nur die Standardnavigation verwenden und auf zugehörige Ereignisse reagieren, sondern auch auf Tastendruck-Ereignisse warten und reagieren.

Nicht alle Geräte und Inhaltstypen unterstützen sämtliche Gerätetasten. Bei einem Gerät mit Zweiwege-Navigation (siehe „Standardnavigationsmodi“ auf Seite 38) generiert Flash Lite z. B. keine Tastendruck-Ereignisse für die Nach-links- und Nach-rechts-Taste. Eine Liste der Tasten und Einzelheiten zu ihrer Verfügbarkeit finden Sie unter „Von Flash Lite unterstützte Tasten“ auf Seite 36.

Schaltflächenereignisse verarbeiten

Mithilfe von Schaltflächen können Sie Ihre Flash Lite-Anwendungen schnell und einfach interaktiv gestalten. Flash Lite unterstützt dieselben Schaltflächenereignisse wie Flash Player auf einem Desktop-Computer, wobei jedoch einige Ereignisse nur für Geräte verfügbar sind, die mit einer Maus oder einem Stift bedient werden (zu diesen Ereignissen zählt beispielsweise `onDragOut`). Bei Geräten, die nur über eine Tastatur bedient werden können, muss eine Schaltfläche den Tastaturfokus haben, bevor sie Ereignisse generieren kann.

Flash Lite unterstützt die folgenden ActionScript-Schaltflächenereignisse:

Schaltflächenereignis	Beschreibung
`onDragOut`	Nur für Geräte mit einer Maus oder einem Stift verfügbar. Wird aufgerufen, wenn der Benutzer mit der Maus auf eine Schaltfläche klickt, die Maustaste gedrückt hält und den Mauszeiger dann von der Schaltfläche weg zieht.
`onDragOver`	Nur für Geräte mit einer Maus oder einem Stift verfügbar. Wird aufgerufen, wenn der Benutzer mit der Maus auf eine andere Stelle klickt, die Maustaste gedrückt hält und den Mauszeiger dann über die Schaltfläche zieht.
`onKeyDown`	Wird aufgerufen, wenn die Schaltfläche den Fokus hat und eine Taste gedrückt wird.
`onKeyUp`	Wird aufgerufen, wenn die Schaltfläche den Fokus hat und eine Taste losgelassen wird.
`onKillFocus`	Wird aufgerufen, wenn der Fokus von einer Schaltfläche entfernt wird.
`onPress`	Wird aufgerufen, wenn der Benutzer die Auswahltaste des Geräts drückt, während die Schaltfläche den Fokus hat. Wird auch aufgerufen, wenn der Benutzer die Maustaste über der Schaltfläche drückt, wenn die Schaltfläche den Fokus hat.
`onRelease`	Wird aufgerufen, wenn der Benutzer die Auswahltaste des Geräts loslässt, während die Schaltfläche den Fokus hat. Wird auch aufgerufen, wenn der Benutzer die Maustaste über der Schaltfläche loslässt, wenn die Schaltfläche den Fokus hat.
`onReleaseOutside`	Wird aufgerufen, wenn die Maustaste losgelassen wird, während der Zeiger sich nicht auf der Schaltfläche befindet, nachdem der Benutzer mit der Maustaste auf die Schaltfläche gedrückt hat.
`onRollOut`	Wird aufgerufen, wenn eine Schaltfläche den Fokus verliert.
`onRollOver`	Wird aufgerufen, wenn eine Schaltfläche den Fokus erhält.
`onSetFocus`	Wird aufgerufen, wenn eine Schaltfläche den Eingabefokus erhält.

Im Folgenden erfahren Sie, wie Sie eine einfache Anwendung erstellen, die Schaltflächenereignisse verarbeitet. Ein Beispiel, wie Sie mit Schaltflächen ein Menü erstellen, finden Sie unter „Einfaches Menü mithilfe von Schaltflächen und Standardnavigation erstellen“ auf Seite 47.

Eine Schaltflächen-Ereignisprozedur erstellen

1 Erstellen Sie in Flash ein neues Mobilgerätdokument, und speichern Sie es unter custom_focus_manager.fla

2 Wählen Sie Fenster > Allgemeine Bibliotheken > Schaltflächen, um eine externe Bibliothek mit vordefinierten Schaltflächensymbolen zu öffnen.

3 Doppelklicken Sie im Bedienfeld „Bibliothek - Schaltflächen“ auf den Ordner „Classic Buttons“ und dann auf den Ordner „Circle Buttons“.

4 Ziehen Sie eine Instanz des Schaltflächensymbols „Menu“ auf die Bühne.

5 Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ den Text **btn_1** ein.

6 Ziehen Sie eine weitere Instanz derselben Schaltfläche auf die Bühne und platzieren Sie sie direkt unter der ersten Schaltfläche.

7 Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ den Text **btn_2** ein.

8 Wählen Sie in der Zeitleiste Bild 1 der Ebene ActionScript aus.

9 Öffnen Sie das Bedienfeld Aktionen (Fenster > Aktionen) und geben Sie folgenden Code ein:

```
// Disable the focus rectangle because buttons have an over state
_focusRect = false;

// Event handlers for btn_1
btn_1.onPress = function() {
    trace("You pressed Button 1");
}
btn_1.onRelease = function() {
    trace("You released Button 1");
}
btn_1.onRollOver = function() {
    trace("Button 1 has focus");
}
btn_1.onRollOut = function() {
    trace("Button 1 lost focus");
}

// Event handlers for btn_2
btn_2.onPress = function() {
    trace("You pressed Button 2");
}
btn_2.onRelease = function() {
    trace("You released Button 2");
}
btn_2.onRollOver = function() {
    trace("Button 2 has focus");
}
btn_2.onRollOut = function() {
    trace("Button 2 lost focus");
}
```

10 <Testen Sie die Anwendung im Emulator (Steuerung > Film testen).

Lesen Sie die Meldungen, die im Bedienfeld „Ausgabe“ angezeigt werden, wenn Sie auf der Emulator-Tastatur die Pfeiltaste „Nach-unten“ und „Nach-oben“ drücken.



Andere Objekttypen unterstützen andere Ereignisse. So enthält das Objekt `TextField` beispielsweise das Ereignis `onChanged`, das aufgerufen wird, wenn sich der Inhalt eines Textfelds ändert. Sie können für diese Ereignisse Ereignisprozedur-Code im selben Format schreiben wie für die Schaltflächen-Ereignisprozeduren in diesem Verfahren. Weitere Informationen zu den Ereignissen, die für Textfelder und Movieclips unterstützt werden, finden Sie unter `TextField` und `MovieClip` im *Flash Lite 2.x und 3 ActionScript-Referenzhandbuch*.

Einfaches Menü mithilfe von Schaltflächen und Standardnavigation erstellen

Zum Erstellen eines einfachen Menüs mithilfe von Schaltflächen und der Standardnavigation müssen Sie das Menü entwerfen und anschließend Ereignisprozedur-Code dafür schreiben. das drei Optionen enthält. Dabei verwenden Sie für jede Menüoption ein Schaltflächensymbol. Dann schreiben Sie Ereignisprozedur-Code, der eine Meldung einblendet, sobald der Benutzer den Mauszeiger über ein Menüelement bewegt (d. h. wenn die zugehörige Schaltfläche den Fokus erhält) und ein Menüelement durch Drücken der Auswahltaste auf dem Gerät auswählt. Weitere Informationen zur Verarbeitung von Schaltflächenereignissen in Flash Lite finden Sie unter „Schaltflächenereignisse verarbeiten“ auf Seite 44.

Beginnen Sie mit einem Flash-Dokument, das bereits teilweise fertig gestellt ist. Wenn Sie ein anderes Gerät oder einen anderen Inhaltstyp verwenden möchten, können Sie diese Einstellungen einfach ändern (siehe „Emulator verwenden“ auf Seite 119).

1 Laden Sie die Datei simple_menu_start.fla, die Sie unter www.adobe.com/go/learn_flt_samples_and_tutorials_de finden, herunter und öffnen Sie sie. Suchen Sie auf dieser Seite mit Beispielen und Tutorials nach der komprimierten Datei für Ihre Flash Lite-Version, laden Sie sie herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf das Beispiel zuzugreifen.

2 Öffnen Sie das Bedienfeld „Bibliothek“ (Fenster > Bibliothek).

   Das Bedienfeld „Bibliothek“ enthält die drei Schaltflächensymbole „News Button“, „Weather Button“ und „Sports Button“.

3 Wählen Sie in der Zeitleiste (Fenster > Zeitleiste) die Ebene Menu Buttons aus.

4 Ziehen Sie eine Instanz des Symbols „News Button“ aus dem Bedienfeld „Bibliothek“ auf die Bühne.

5 Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ den Text **btn_news** ein.

6 Wiederholen Sie die Schritte 4 und 5 für die Sports- und Weather-Schaltflächen und nennen Sie sie **btn_sports** bzw. **btn_weather**.

7 Richten Sie die drei Schaltflächen vertikal aus, wie im folgenden Beispiel gezeigt:



8 Wählen Sie im Bedienfeld „Werkzeuge“ das Textwerkzeug aus und erstellen Sie im unteren Bühnenbereich ein Textfeld.

   In diesem Textfeld wird eine kurze Meldung angezeigt, wenn der Benutzer den Mauszeiger über ein Menüelement bewegt.

9 Vergewissern Sie sich, dass das neue Textfeld noch ausgewählt ist und führen Sie im Eigenschafteninspektor die folgenden Schritte aus:

   a Wählen Sie im Popupmenü „Texttyp“ die Option „Dynamischer Text“ aus.

b Geben Sie **txt_status** in das Textfeld „Instanzname“ ein.

Nun sollte die Bühne ungefähr wie in der folgenden Abbildung aussehen:



10 Wählen Sie in der Zeitleiste Bild 1 der Ebene ActionScript aus.

11 Öffnen Sie das Bedienfeld Aktionen (Fenster > Aktionen) und geben Sie folgenden Code ein:

```
// Disable the focus rectangle because buttons have an over state
_focusRect = false;

btn_news.onRollOver = function() {
    txt_status.text = "Press to select News";
}
btn_news.onPress = function() {
    txt_status.text = "You selected News";
}
btn_sports.onRollOver = function() {
    txt_status.text = "Press to select Sports";
}
btn_sports.onPress = function() {
    txt_status.text = "You selected Sports";
}
btn_weather.onRollOver = function() {
    txt_status.text = "Press to select Weather";
}
btn_weather.onPress = function() {
    txt_status.text = "You selected Weather";
}
```

12 Wählen Sie Steuerung > Film testen, um eine Vorschau der Anwendung im Emulator anzuzeigen.

Klicken Sie im Emulator mit der Maus auf die abgebildete Taste <Nach-unten > (oder drücken Sie auf Ihrer Computertastatur die Taste <Nach-unten >), um zwischen den Menüoptionen hin und her zu wechseln. Zum Auswählen eines Menüelements klicken Sie mit der Maus auf die Auswahltaste des Emulators (oder drücken Sie die Eingabetaste auf der Computertastatur).



Tastenereignisse verarbeiten

Flash Lite generiert Tastendruck-Ereignisse, wenn der Benutzer Tasten auf dem Gerät drückt. Sie können Ereignisprozedur-Code zur Reaktion auf diese Schaltflächenereignisse schreiben. Eine Liste der von Flash Lite unterstützten Tasten finden Sie unter „Von Flash Lite unterstützte Tasten“ auf Seite 36.

In der folgenden Tabelle werden häufige Gerätetasten und die zugehörigen ActionScript-Tastencodes sowie die Tastencode-Konstanten für diese Tasten aufgelistet:

Gerätetaste	ActionScript-Tastencode/Tastencode-Konstante
Auswahltaste	Key.ENTER
Nach-oben-Navigationstaste	Key.UP
Nach-unten-Navigationstaste	Key.DOWN
Nach-links-Navigationstaste	Key.LEFT
Nach-rechts-Navigationstaste	Key.RIGHT
Linke Softtaste	ExtendedKey.SOFT1 (oder soft1)
Rechte Softtaste	ExtendedKey.SOFT2 (oder soft2)
0	48
1	49

Gerätetaste	ActionScript-Tastencode/Tastencode-Konstante
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57
*	56
#	51

Ereignis-Listener schreiben

Über Ereignis-Listener können so genannte *Listener-Objekte* Ereignisse empfangen, die von anderen Objekten, so genannten *Broadcaster-Objekten*, gesendet wurden. Mit dem Broadcaster-Objekt wird das Listener-Objekt zum Empfangen von Ereignissen vom Broadcaster registriert. Weitere Informationen finden Sie unter „Ereignis-Listener verwenden“ im *ActionScript 2.x in Flash – Arbeitshandbuch*.

Bei einem einfachen Verfahren zur Verarbeitung von Tastendruck-Ereignissen erstellen Sie ein Tasten-Listener-Objekt, das eine Funktion `onKeyDown` oder `onKeyUp` definiert. Dann registrieren Sie dieses Objekt mit der Methode `Key.addListener()`. Mit dem folgenden Beispielcode wird ein Tasten-Listener definiert, der reagiert, wenn der Benutzer die Nach-rechts-Taste auf dem Gerät drückt:

```
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    if (Key.getCode() == Key.RIGHT) {
        trace("You pressed the right arrow key");
    }
}
Key.addListener(myListener);
```

Tastendruck-Ereignisse mithilfe eines Tasten-Listeners verarbeiten

1 Erstellen Sie in Flash ein neues Mobilgerätdokument, und speichern Sie es unter keylistener.fla.

2 Wählen Sie in der Zeitleiste die Ebene „Content“ aus.

3 Erstellen Sie mit dem Ellipsenwerkzeug auf der Bühne eine Ellipse oder einen Kreis. Wandeln Sie die Ellipse oder den Kreis in einen Movieclip um.

4 Achten Sie darauf, dass der neue Movieclip ausgewählt ist. Geben Sie dann im Eigenschafteninspektor im Textfeld „Instanzname“ den Namen **circle** ein.

5 Wählen Sie in der Zeitleiste Bild 1 in der Ebene 1 aus.

6 Öffnen Sie das Bedienfeld Aktionen (Fenster > Aktionen) und geben Sie folgenden Code ein:

```
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    if (Key.getCode() == Key.LEFT) {
        circle._x -= 10;
    } else if (Key.getCode() == Key.RIGHT) {
        circle._x += 10;
    } else if (Key.getCode() == Key.UP) {
        circle._y -= 10;
    } else if (Key.getCode() == Key.DOWN) {
        circle._y += 10;
    }
};
Key.addListener(myListener);
```

7 Wählen Sie Steuerung > Film testen, um die Anwendung zu testen.

Drücken Sie die vier Navigationstasten auf der Emulator-Tastatur (oder die entsprechenden Pfeiltasten auf Ihrer Tastatur), um den Kreis auf der Bühne zu verschieben.

Softtasten verwenden

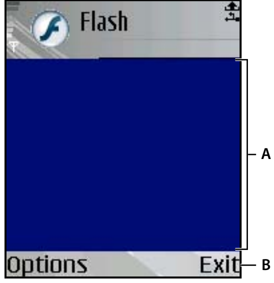
Zur Verwendung der Softkeys in Ihrer Flash Lite-Anwendung müssen Sie zunächst den Befehl `SetSoftKeys` aufrufen. Danach generiert Flash Lite ein `ExtendedKey.SOFT1`-Ereignis, wenn der Benutzer die linke Softtaste drückt, und ein `ExtendedKey.SOFT2`-Ereignis, wenn der Benutzer die rechte Softtaste drückt. Sie müssen in ActionScript Ereignisprozedur-Code schreiben, der auf diese Ereignisse reagiert und die gewünschten Aktionen einleitet.

Der Befehl `SetSoftKeys` hat zwei Parameter für die Beschriftung der linken und der rechten Softtaste. Diese Beschriftungen werden angezeigt, wenn eine Anwendung *nicht* im Vollbildmodus ausgeführt wird. Bei Anwendungen, die im Vollbildmodus ausgeführt werden, sind diese Beschriftungen nicht zu sehen; Sie müssen daher eigene Beschriftungen erstellen und diese auf der Bühne an den Stellen platzieren, an denen sich die Softtasten befinden.

Das folgende Codebeispiel enthält einen `SetSoftKeys`-Befehlsaufruf:

```
fscommand2("SetSoftKeys", "Options", "Exit");
```

Im folgenden Beispiel sehen Sie die Auswirkungen dieses Befehls auf eine Anwendung, die auf einem Gerät im Normalmodus (nicht im Vollbildmodus) ausgeführt wird:



***A.** Verfügbarer Bildschirmbereich bei Anwendungen, die nicht im Vollbildmodus ausgeführt werden* ***B.** Vom Gerät angezeigte Beschriftungen der Softtasten*

Wenn Sie den Vollbildmodus aktivieren, d. h., wenn Sie `fscommand("fullscreen", true)` aufrufen, sind die Bezeichnungen, die Sie als Parameter für den Befehl `SetSoftKeys` festlegen, nicht sichtbar. Sie müssen demnach für Anwendungen im Vollbildmodus eigene Beschriftungen für die Softkeys erstellen, wie in der folgenden Abbildung veranschaulicht:



Benutzerdefinierte Beschriftungen für Softkeys

Weitere Informationen zum Befehl `SetSoftKeys` finden Sie im Eintrag zur `fscommand2-Funktion` im *Flash Lite 2.x ActionScript Language Reference*.

Softtasten in einer Anwendung verwenden

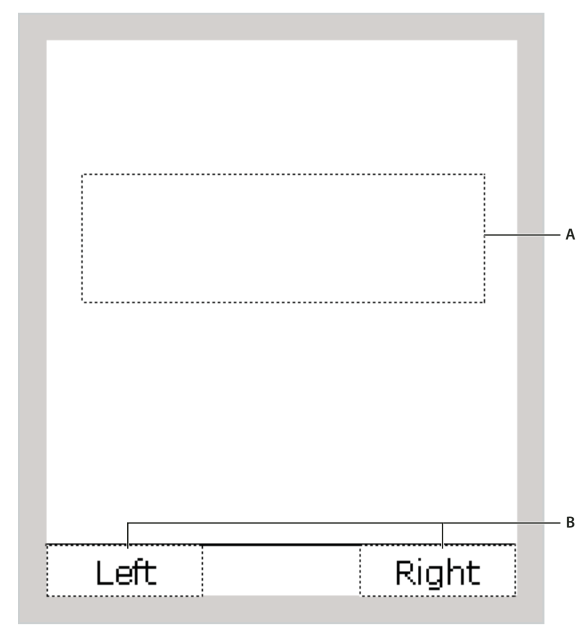
1 Erstellen Sie in Flash ein neues Mobilgerätdokument, und speichern Sie es unter softkey.fla.

2 Erstellen Sie mit dem Textwerkzeug ein statisches Textfeld namens „Left“ und platzieren Sie es über der linken Softtaste des Geräts unten links auf der Bühne.

3 Erstellen Sie ein zweites statisches Textfeld namens „Right“ und platzieren Sie es über der rechten Softtaste des Geräts unten rechts auf der Bühne.

4 Erstellen Sie mit dem Textwerkzeug ein dynamisches Textfeld in der Mitte der Bühne.

Die Bühne des Dokuments sollte etwa folgendermaßen aussehen:



A. *Dynamisches Textfeld* ***B.*** *Beschriftungen der Softtasten*

5 Vergewissern Sie sich, dass das dynamische Textfeld noch ausgewählt ist. Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ den Text **status** ein.

6 Öffnen Sie das Bedienfeld Aktionen (Fenster > Aktionen) und wählen Sie in der Zeitleiste Bild 1 in der Ebene 1 aus.

7 Geben Sie im Bedienfeld „Aktionen“ folgenden Code ein:

```
fscommand2("SetSoftKeys", "Left", "Right");
fscommand2("FullScreen", true);
```

8 Erstellen und registrieren Sie ein Objekt als Reaktion auf Tastendruck-Ereignisse (siehe „Tastendruck-Ereignisse mithilfe eines Tasten-Listeners verarbeiten“ auf Seite 50), indem Sie den folgenden Code in das Bedienfeld „Aktionen“ eingeben:

```
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    if (Key.getCode() == ExtendedKey.SOFT1) {
        // Handle left soft keypress event.
        status.text = "You pressed the Left soft key.";

    } else if (Key.getCode() == ExtendedKey.SOFT2) {
        // Handle right soft keypress event.
        status.text = "You pressed the Right soft key.";
    }
};
Key.addListener(myListener);
```

9 Wählen Sie Steuerung > Film testen, um die Anwendung im Emulator zu testen.

Klicken Sie zum Testen der Anwendung mit der Maus auf die linke und rechte Softtaste im Emulator, oder drücken Sie die Nach-oben- bzw. Nach-unten-Taste auf der Computertastatur.



Kapitel 5: Text und Schriftarten verwenden

Sie können Anwendungen, die mit Macromedia Flash Lite 2.x von Adobe und Adobe Flash Lite 3.x entwickelt wurden, statische und dynamische Textfelder und Eingabetextfelder hinzufügen.

Text

Text in Flash Lite

Flash Lite 2.x und 3.x unterstützen die folgenden Textfunktionen:

- Statische, dynamische und Eingabetextfelder

  Der Inhalt eines statischen Textfelds kann sich zur Laufzeit nicht ändern, der Inhalt von dynamischen und Eingabetextfeldern hingegen schon. Texteingabefelder dienen zur Eingabe von Text durch den Anwender. Flash Lite 2.x und 3.x unterstützen die Inline-Texteingabe bei den meisten Geräten. Bei Geräten, die komplexe Sprachen unterstützten und unter Flash Lite 2.0 nutzen die Eingabetextfelder den generische Texteingabe-Mechanismus des Gerätes. Weitere Informationen zu Eingabetextfeldern finden Sie unter „Eingabetextfelder verwenden“ auf Seite 55.

- Eingebettete Schriftarten und Geräteschriftarten

  Textfelder können in Flash Lite mit Schriftarten dargestellt werden, die entweder als Konturen in der SWF-Datei eingebettet oder im Gerät verfügbar sind. Weitere Informationen über Methoden zur Schriftwiedergabe finden Sie unter „Methoden zur Schriftwiedergabe in Flash Lite“ auf Seite 64.

- Unicode-Textkodierung

  Flash Lite kann Text in jeder Sprache anzeigen, für die Schriftarten mit den erforderlichen Zeichen verfügbar sind. Weitere Informationen zum mehrsprachigen Authoring in Flash finden Sie unter „Text in mehreren Sprachen erstellen“ im Handbuch *Using Flash*.

- Teilunterstützung für HTML-Formatierung und die Eigenschaften der TextFormat-Klasse
- Text mit Bildlauf

Flash Lite unterstützt nicht alle Textfunktionen, die in der Desktopversion von Flash Player verfügbar sind. In Flash Lite gelten die folgenden Einschränkungen:

- Das erweiterte Anti-Aliasing, die verbesserte Schriftwiedergabetechnologie in Adobe Flash Player ab Version 8 von Adobe, wird nicht unterstützt.
- Das Formatieren von Text wird zwar unterstützt, doch nur die Optionen für Farbe, Schriftart, Schriftgröße, Fett- und Kursivdarstellung sind verfügbar. Außerdem wird die Formatierung nicht angezeigt, wenn die Gerätetextschriftart die ausgewählte Option nicht beinhaltet. So wird ein kursiv formatiertes Feld beispielsweise als regulärer Text angezeigt, wenn die Geräteschriftart nicht in Kursivausführung zur Verfügung steht.
- Gerätetext kann nicht maskiert, als Maske verwendet oder mit Transparenz dargestellt werden.

- Die Formatierungsoption „Text als HTML wiedergeben“ wird für Eingabetextfelder und dynamische Textfelder teilweise unterstützt. HTML-Tags werden im Text nicht als solche angezeigt, sie bleiben jedoch wirkungslos, mit Ausnahme der folgenden Tags: `p`, `br`, `sbr`, `font` (mit den Attributen `face`, `color` und `size`), `b` und `i`.
- Flash Lite bietet keine Unterstützung für CSS (Cascading Style Sheets).
- Flash-Komponenten wie Label, TextArea und TextInput werden nicht unterstützt.
- Die TextField- und TextFormat-Objekte werden teilweise unterstützt. Für Arabisch, Hebräisch und Thai gelten zusätzliche Einschränkungen. Weitere Informationen finden Sie im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.
- Die XML- und XMLNode-Objekte werden unterstützt.

Flash Lite 2.1 und später bieten Unterstützung für die Inline-Texteingabe, die automatische Texterkennung und XMLSocket.

- Durch die Unterstützung der Inline-Texteingabe kann der Anwender Text direkt in die Textfelder eingeben.
- Die automatische Texterkennung ermöglicht Funktionen wie z. B. Wortvervollständigung und Kandidatenlisten. Flash Lite 2.1 und später unterstützen auf allen Plattformen die gängigsten Programme zur automatischen Texterkennung (wie T9, eZiTap/eZiText und iTap), vorausgesetzt sie sind auf ähnliche Weise umgesetzt wie die Standard-APIs, die von den Herstellern der automatischen Texterkennungsprogramme angeboten werden.
- Durch XMLSocket wird der Flash Desktop auch in Flash Lite unterstützt. Jetzt können Softwareentwickler ständige Datenübertragungen mit geringen Latenzzeiten für Anwendungen wie Spiele und Chatprogramme erstellen.

Text erstellen und formatieren

Text wird in Flash Lite genauso erstellt und formatiert wie in einer Flash-Desktop-Anwendung.

Weitere Informationen zur Verwendung von Text in Flash finden Sie unter den folgenden Themen im Handbuch „Flash verwenden“:

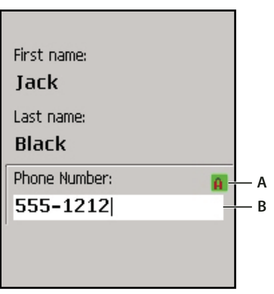
- Text erstellen
- Textattribute festlegen
- Text bearbeiten
- Text mit ActionScript steuern

Eine Liste der Textfunktionen, die in Flash Lite nicht unterstützt werden, finden Sie unter „Text in Flash Lite“ auf Seite 54.

Texteingabe

Eingabetextfelder verwenden

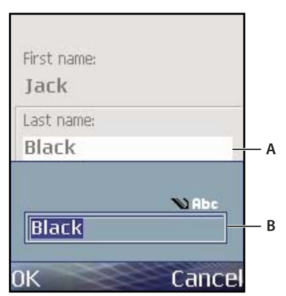
Flash Lite 2.1 und später unterstützen die Inline-Texteingabe. Mit dieser Funktion können Benutzer Textfelder direkt in der Flash Lite-Anwendung bearbeiten und müssen nicht mehr wie in früheren Versionen in einem separaten Texteingabefeld arbeiten. In der folgenden Abbildung wird ein Inline-Texteingabefeld auf einem Nokia-Gerät mit der Benutzeroberfläche Series 60 gezeigt:



***A.** Eingabemodussymbol* ***B.** Inline bearbeitetes Textfeld*

Flash Lite 2.1 und später unterstützt die Inline-Texteingabe bei lateinischen und asiatischen Sprachen, jedoch nicht bei komplexen Sprachen wie Arabisch, Hebräisch, Urdu, Farsi und einige asiatischen Sprachen.

Bei Geräten, die Flash Lite 2.0 und Flash Lite 1.1 ausführen und im Adobe Device Central-Emulator für alle Versionen bearbeiten Benutzer den Inhalt von Eingabetextfeldern in einem modalen Dialogfeld, das über dem Flash Lite-Inhalt angezeigt wird. In der folgenden Abbildung wird ein Texteingabe-Dialogfeld auf einem Symbian™-Gerät mit der Benutzeroberfläche Series 60 gezeigt, auf dem Flash Lite 2.0 ausgeführt wird:



***A.** Texteingabe-Dialogfeld* ***B.** Bearbeitetes Textfeld*

Im Allgemeinen funktionieren bestehende Flash Lite 2.0- und Flash Lite 1.1-Anwendungen auch in Flash Lite 2.1 und später, ohne dass Änderungen erforderlich werden. Hier wird der Text statt in ein modales Dialogfeld einfach inline in ein Eingabetextfeld eingegeben. Ältere Inhalte müssen jedoch neu erstellt werden, um auch die Funktionen nutzen zu können, die nur in Flash Lite 2.1 und später verfügbar sind – z. B. die Möglichkeit, Textauswahl einzustellen oder der Befehl `activateTextField`.

Weitere Informationen zum Eingabetext-Dialogfeld finden Sie unter „Texteingabe-Dialogfeld des Geräts (Flash Lite 2.0)“ auf Seite 59.

Inline-Texteingabe verwenden (Flash Lite 2.1 und später)

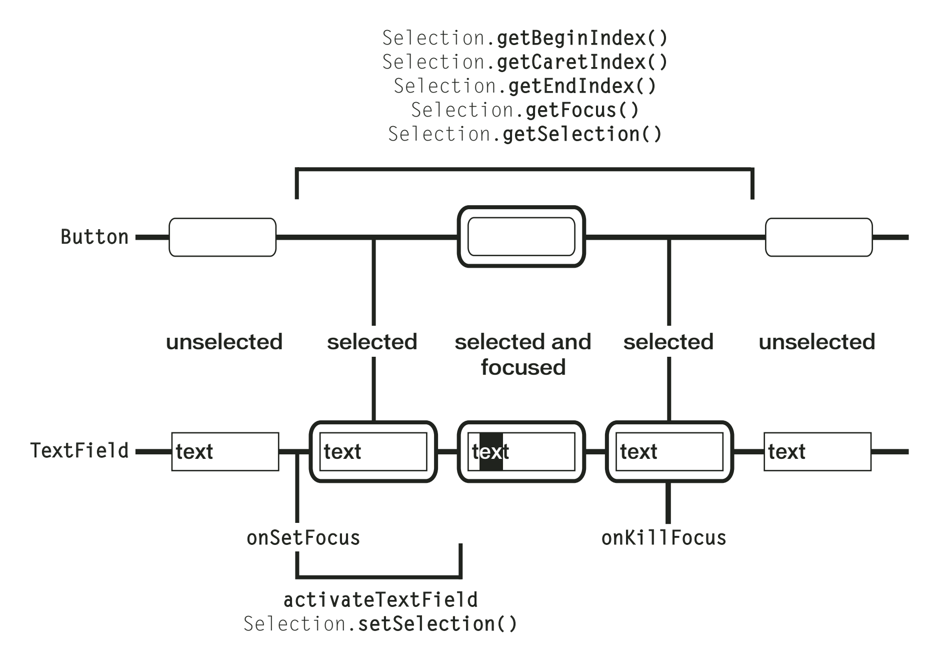
Flash Lite 2.1 und später unterstützen die Inline-Texteingabe, die es Benutzern ermöglicht, Text direkt in ein Eingabetextfeld einzugeben bzw. darin zu bearbeiten. Bei der Eingabe von Text in eine Flash Lite-Anwendung arbeiten die Benutzer mit dem geräteeigenen Eingabemethodeneditor (Input Method Editor / IME) – der Flash Lite Player verarbeitet die Benutzereingabe nicht selbst.

***Hinweis:** Der Flash Lite-Emulator von Adobe Device Central zeigt keinen Inline-Text an. Weitere Informationen finden Sie unter „Inline-Text testen (Flash Lite 2.1). und höher“ auf Seite 119*

Solange ein Eingabetextfeld aktiv ist, wird der Flash Lite 2.1 und 3.x Player im *eingeschränkten* Modus ausgeführt. In diesem Modus werden alle Tastenereignisse vom geräteeigenen IME verarbeitet und es wird kein ActionScript-Code ausgeführt. Darüber hinaus werden alle Ereignisse zum Anhalten der Wiedergabe von Animationen, Sounds und Video ignoriert. Nachdem ein Textfeld deaktiviert wurde, kehrt Flash Lite zum normalen Betriebsmodus zurück.

Bevor ein Textfeld eine Texteingabe von einem Benutzer akzeptieren kann, muss es *aktiviert* werden. Ein Benutzer kann ein Texteingabefeld aktivieren, indem er zunächst den Auswahlfokus auf das Feld verschiebt und dann die Auswahltaste des Geräts oder eine Softtaste drückt, die bewirkt, dass die Anwendung den Fokus auf die Texteingabe setzt. (Weitere Informationen zum Aktivieren eines Textfelds finden Sie unter „Eingabetextfelder mit ActionScript aktivieren“ auf Seite 57.

In der folgenden Abbildung werden die unterschiedlichen Zustände des Eingabetextfeldes angezeigt: nicht ausgewählt, ausgewählt (aber nicht aktiviert) und aktiviert:



Inline-Eingabetextfelder und Navigation

Ein aktiviertes Textfeld enthält einen Zeiger, der den aktuellen Einfügepunkt kennzeichnet. Der Benutzer kann die Position des Zeigers innerhalb des Textfeldes mithilfe der Navigationstasten des Gerätes ändern.

In der Standardeinstellung wird in Flash Lite das Textfeld, das momentan den Fokus hat, mit einem Fokusrechteck gekennzeichnet. Das Fokusrechteck kann manchmal den blinkenden Zeiger oder das Einfügesymbol in einem aktivierten Textfeld verdecken. Daher sollten Sie das Fokusrechteck deaktivieren (indem Sie die Eigenschaft `_focusRect` auf `false` setzen) und stattdessen ein benutzerdefiniertes Fokussymbol verwenden. Ein Beispiel einer Anwendung, die ein benutzerdefiniertes Fokussymbol verwendet, finden Sie auf der Flash Lite-Website auf der Seite mit Beispielen und Tutorials unter www.adobe.com/go/learn_flt_samples_and_tutorials_de. Laden Sie die komprimierte Datei (ZIP) für Ihre ActionScript-Version herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf das Beispiel zuzugreifen.

Eingabetextfelder mit ActionScript aktivieren

Ein Eingabetextfeld, das den Tastaturfokus besitzt, kann durch Drücken der Auswahltaste des Geräts aktiviert werden. Eine Flash Lite-Anwendung kann ein Eingabetextfeld auch automatisch aktivieren, wenn sie den Fokus über den Befehl `activateTextField` erhält. Dieser Befehl aktiviert das momentan ausgewählte Textfeld. Falls bei Ausführung des Befehls kein Feld ausgewählt ist, geschieht nichts.

In der Regel wird der Befehl `activateTextField` aus einer `Selection.onSetFocus`-Prozedur oder einer `TextField.onSetFocus`-Prozedur heraus aufgerufen. Angenommen, Ihre Anwendung enthält zwei (oder mehr) Eingabetextfelder auf der Bühne. Der folgende Code aktiviert automatisch das Textfeld, das den Fokus erhält:

```
var focusListener:Object = new Object ();
focusListener.onSetFocus = function (oldFocus, newFocus) {
    // Call activateTextField:
    fscommand ("activateTextField", "");
};
TextField1.addListener (focusListener);
```

Sie können auch die Prozedur `TextField.prototype.onSetFocus` verwenden, um alle Textfelder zu aktivieren, wenn sie den Fokus erhalten.

Es ist auch möglich, andere Tasten als die Auswahltaste des Gerätes zum Auslösen des Befehls `activateTextfield` verwenden. Der folgende Code aktiviert ein Textfeld für alle Zahlentasten, sodass es einfacher wird, beispielsweise den Buchstaben „a“ einzugeben. Wenn die Anwendung den Befehl `activateTextfield` in der Prozedur `TextField.onSetFocus` enthält, müsste der Benutzer zunächst die Auswahltaste und dann 2 drücken. Mit dem folgenden Code muss der Benutzer jedoch nur zweimal die Taste `2` drücken, eine wesentlich intuitivere Lösung.

```
var keyListener:Object = new Object();
keyListener.onKeyDown = function() {
    if (Key.getCode() >= 48 && Key.getCode() <= 57 ){
        fscommand("activateTextField", "");
    }
Selection.addListener (keyListener);
};
```

Eine vollständige Beispielanwendung, die diese Technik verwendet, finden Sie im Eingabebeispiel für Inline-Text unter www.adobe.com/go/learn_flt_samples_and_tutorials_de. Laden Sie die komprimierte Datei (ZIP) für Ihre ActionScript-Version herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf das Beispiel zuzugreifen.

Eingabemodussymbol

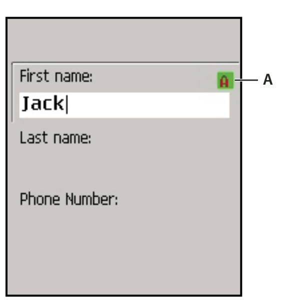
Die meisten Geräte bieten verschiedene Eingabemodi wie z. B. automatische Texterkennung oder manuelle Texteingabe (*Triple Tap* oder *Multi Tap*) oder die Nur-Zahlen-Modi an, um Benutzer bei der allgemeinen Texteingabe zu unterstützen.

Wenn Flash Lite im Vollbildmodus ausgeführt wird, zeigt das Gerät den Buchstaben „A“ für den alphanumerischen Eingabemodus und das Nummernzeichen (#) für den numerischen Eingabemodus an. Wird Flash Lite nicht im Vollbildmodus ausgeführt, zeigt das Gerät eventuell ein Eingabemodussymbol in der Statusleiste oder an einer anderen Position auf dem Bildschirm an. In der folgenden Abbildung wird das Eingabemodussymbol in einer nicht im Vollbildmodus dargestellten eigenständigen Version von Flash Lite 2.1 auf einem Gerät mit der Benutzeroberfläche Series 60 angezeigt:



A. *Eingabemodussymbol, nicht im Vollbildmodus* ***B.*** *Bearbeitetes Textfeld*

Wird der Player im Vollbildmodus ausgeführt, kann das Gerät ein Eingabemodussymbol an einer beliebigen Position auf dem Bildschirm anzeigen. In der folgenden Abbildung wird das Eingabemodussymbol in einer im Vollbildmodus dargestellten eigenständigen Version von Flash Lite 2.1 auf einem Gerät mit der Benutzeroberfläche Series 60 angezeigt:



Eingabemodussymbol, Vollbildmodus

Das in der obigen Abbildung dargestellte Eingabemodussymbol für Anwendungen im Vollbildmodus ist eine Referenzimplementierung für den eigenständigen Player auf Geräten mit der Benutzeroberfläche Series 60. Das jeweilige Symbol, sofern vorhanden, wird durch das Gerät vorgegeben.

Texteingabe-Dialogfeld des Geräts (Flash Lite 2.0)

Um das Texteingabe-Dialogfeld des Geräts zu öffnen, muss der Benutzer zuerst einem Eingabetextfeld den Fokus zuweisen und dann die Auswahltaste des Geräts drücken.

Das Texteingabe-Dialogfeld ist modal, d. h. der Benutzer kann den Flash-Inhalt nicht ändern, solange das Dialogfeld den Fokus hat. Außerdem wird der Abspielkopf in der Flash-Anwendung von Flash Lite angehalten, während das Dialogfeld den Fokus hat.

Wenn der Benutzer „OK“ auswählt (linke Softtaste), wird das Texteingabe-Dialogfeld geschlossen und Flash Lite weist automatisch den Text dem Eingabetextfeld zu. Wenn der Benutzer „Abbrechen“ auswählt (rechte Softtaste), wird dem Eingabetextfeld kein Text zugewiesen.

Der Adobe Device Central-Emulator imitiert die Funktionen des Texteingabe-Dialogfelds, wenn Sie eine Anwendung im Flash-Authoring-Tool testen. In der folgenden Abbildung sehen Sie das Texteingabe-Dialogfeld im Emulator:



Texteingabe-Dialogfeld im Emulator

Ein Beispiel für die Verwendung eines Eingabetextfelds in einer Anwendung finden Sie unter „Beispielanwendung mit verschiedenen Textfeldern erstellen“ auf Seite 62.

Typen von Eingabetextfeldern festlegen

Flash Lite unterstützt einzeilige, mehrzeilige und kennwortspezifische Eingabetextfelder. Den Typ eines Eingabetextfelds legen Sie im Eigenschafteninspektor im Popupmenü „Zeilentyp“ fest, wie Sie in der folgenden Abbildung sehen:



Der für ein Eingabetextfeld festgelegte Zeilentyp bestimmt, wie sich das Texteingabe-Dialogfeld des Geräts verhält, wenn ein Benutzer das Textfeld bearbeitet.

Wenn ein Benutzer z. B. ein einzeiliges Eingabetextfeld bearbeitet, wird im Texteingabe-Dialogfeld des Geräts ein einzeiliges Textfeld angezeigt. Für den Text in diesem Textfeld wird ein horizontaler Bildlauf ausgeführt, wenn der Benutzer mehr Zeichen eingibt, als angezeigt werden können.

Zeicheneingabe beschränken

Mit dem Befehl `SetInputTextType` können Sie festlegen, welche Zeichen der Benutzer im Texteingabe-Dialogfeld eingeben kann. Angenommen, eine Anwendung enthält ein Eingabetextfeld, in das der Benutzer einen numerischen Wert eingeben soll – z. B. sein Alter. Das Eingabetextfeld in diesem Beispiel hat den Variablennamen `ageVar`. Damit der Benutzer nur numerische Werte in das Texteingabe-Dialogfeld eingeben kann, können Sie der Anwendung folgenden Code hinzufügen:

```
fscommand2("SetInputTextType", "ageVar", "Numeric");
```

Öffnet ein Benutzer dieses Texteingabe-Dialogfeld, so kann er ausschließlich numerische Werte in die Textfelder eingeben.

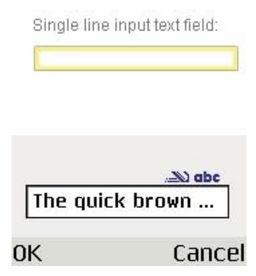
Weitere Informationen hierzu finden Sie unter `SetInputTextType` im Eintrag zur `fscommand2`-Funktion im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch.*

Eingabetext-Dialogfelder (Flash Lite 2.0)

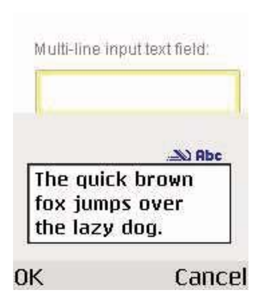
In Flash Lite 2.0 gibt der Benutzer Text nicht direkt in ein Onscreen-Textfeld, sondern in ein separates modales Dialogfeld ein (das von der Hostanwendung und nicht von Flash Lite gesteuert wird). Während dieser Interaktion wird der Flash Lite Player unterbrochen, bis der Benutzer das Dialogfeld schließt.

(In Flash Lite 2.1 und später kann Text direkt in ein Onscreen-Textfeld eingegeben werden.)

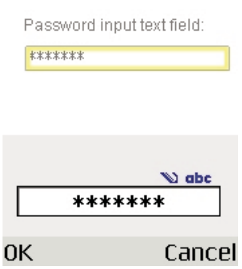
In der folgenden Abbildung sehen Sie das für ein einzeiliges Eingabetextfeld angezeigte Texteingabe-Dialogfeld eines Geräts in einer Flash Lite 2.0-Anwendung:



Wenn ein Benutzer ein mehrzeiliges Eingabetextfeld bearbeitet, wird das Texteingabe-Dialogfeld des Geräts bei Bedarf erweitert, um den gesamten vom Benutzer eingegebenen Text aufzunehmen:



Wenn ein Benutzer ein Eingabetextfeld für ein Kennwort bearbeitet, werden im Texteingabe-Dialogfeld des Geräts alle vom Benutzer eingegebenen Zeichen angezeigt. Erst wenn der Benutzer auf OK klickt, werden alle Zeichen des Kennworts durch Sternchen ersetzt. Dies wird in der folgenden Abbildung verdeutlicht:



Beispielanwendung mit verschiedenen Textfeldern erstellen

Die folgende einfache Anwendung erhält Texteingaben vom Benutzer. Dieser Text wird dann in einem HTML-fähigen dynamischen Textfeld formatiert und angezeigt. In der Anwendung wird auch der Befehl `SetFocusRectColor` verwendet, um die Standardfarbe des Fokusrechtecks (Gelb) in Schwarz zu ändern.

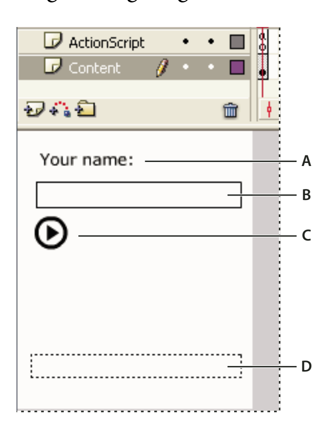
Eine vollständige Beispielanwendung (textfield_example.fla), die diese Technik verwendet, finden Sie unter www.adobe.com/go/learn_flt_samples_and_tutorials_de. Laden Sie die komprimierte Datei (ZIP) für Ihre ActionScript-Version herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“, um auf das Beispiel zuzugreifen.

1 Erstellen Sie in Flash ein neues Mobilgerätdokument, und speichern Sie es unter textfield.fla.

2 Wählen Sie im Bedienfeld „Werkzeuge“ das Textwerkzeug aus und erstellen Sie ein einzeiliges Textfeld im oberen Bereich der Bühne.

3 Achten Sie darauf, dass das Textfeld ausgewählt ist. Wählen Sie dann im Eigenschafteninspektor im Popupmenü „Texttyp“ die Option „Eingabetext“ und im Popupmenü „Methode zur Schriftwiedergabe“ die Option „Geräteschriftarten verwenden“ aus und geben Sie **inputTxt** in das Textfeld „Instanzname“ ein.

4 Erstellen Sie unterhalb des ersten Textfelds ein weiteres Textfeld, das mehrmals so hoch ist wie das erste, wie im Folgenden gezeigt:



5 Achten Sie darauf, dass das zweite Textfeld ausgewählt ist und wählen Sie dann im Eigenschafteninspektor die folgenden Optionen aus: im Popupmenü „Texttyp“ die Option „Dynamischer Text“, im Popupmenü „Zeilentyp“ die Option „Mehrzeilig“, die Option „Text als HTML wiedergeben“ und im Popupmenü „Methode zur Schriftwiedergabe“ die Option „Geräteschriftarten verwenden“. Geben Sie dann **messageTxt** in das Textfeld „Instanzname“ ein.

6 Wählen Sie in der Zeitleiste Bild 1 in der Ebene 1 aus.

7 Öffnen Sie das Bedienfeld „Aktionen“ („Fenster“ > „Aktionen“) und geben Sie folgenden Code ein:

```
Selection.setFocus(inputTxt);
fscommand2("SetFocusRectColor", 0, 0, 0);
inputTxt.onChanged = function() {
    messageTxt.htmlText = "You entered: <i>" + this.text + "</i>";
}
```

Die Methode `Selection.setFocus()` legt den ursprünglichen Fokus auf das Eingabetextfeld (`inputTxt`). Dann wird mit dem Funktionsaufruf `fscommand2()` eine benutzerdefinierte Farbe für das Fokusrechteck festgelegt. Schließlich bewirkt die Ereignisprozedur `onChanged` des Eingabetextfelds, die bei jeder Änderung des Inhalts aufgerufen wird, dass der vom Benutzer im Textfeld `messageTxt` eingegebene Text formatiert und angezeigt wird.

8 Speichern Sie Ihre Änderungen und testen Sie die Anwendung im Emulator (Steuerung > „Film testen“).

9 Drücken Sie die Auswahltaste des Emulators, um das Dialogfeld für die Texteingabe zu öffnen und geben Sie über die Computertastatur einen Text ein. Klicken Sie auf „OK“, um das Dialogfeld zu schließen. Der eingegebene Text wird im Textfeld `messageTxt` in Kursivschrift angezeigt.

Schriftwiedergabe

Methoden zur Schriftwiedergabe in Flash Lite

In Flash Lite können die in Textfeldern verwendeten Schriftarten folgendermaßen wiedergegeben werden:

Auf dem Gerät verfügbare Schriftarten verwenden Sie können einem Textfeld eine Schriftart zuweisen, von der Sie wissen, dass sie auf dem Gerät verfügbar ist. Sie können aber auch eine der drei generischen Geräteschriftarten angeben (_sans, _serif oder _typewriter), die im Popupmenü „Schriftart" zur Auswahl stehen. Wenn Sie eine generische Geräteschriftart auswählen, versucht Flash Lite zur Laufzeit, die generische Schriftart einer Schriftart auf dem Gerät zuzuordnen (z. B. wird _sans einer serifenlosen Schriftart zugeordnet, sofern vorhanden).

Schriftart als Bitmap wiedergeben Zur Wiedergabe von Bitmaptext richtet Flash Lite die Schriftkonturen an Pixelbegrenzungen aus. Dadurch wird die Lesbarkeit von Text bei kleineren Schriftgrößen erhöht (z. B. bei 10 Punkt oder weniger). Diese Option setzt voraus, dass Sie die Schriftkonturen der ausgewählten Schriftart in die veröffentlichte SWF-Datei einbetten. „Schriftkonturen in SWF-Dateien einbetten" auf Seite 66

Schriftart als Vektoren mit Anti-Aliasing wiedergeben Anti-Alias-Text wird in Flash Lite mithilfe von vektorbasierten Darstellungen der Schriftkonturen wiedergegeben, die Sie in die veröffentlichte SWF-Datei einbetten. „Schriftkonturen in SWF-Dateien einbetten" auf Seite 66

Wenn Sie einem Textfeld eine Methode zur Schriftwiedergabe zuweisen möchten, wählen Sie im Eigenschafteninspektor im entsprechenden Popupmenü eine Option aus. Im Popupmenü „Methode zur Schriftwiedergabe" stehen fünf Optionen zur Auswahl, von denen jedoch nur drei für Flash Lite-Entwickler verfügbar sind. Die beiden anderen Methoden („Anti-Aliasing für Lesbarkeit" und „Benutzerdefiniertes Anti-Alias") stehen nur für Anwendungen zur Verfügung, die für Flash Player 8 oder höhere Versionen auf Desktop-Computern vorgesehen sind.

Beachten Sie bei der Wahl zwischen Anti-Alias, Bitmaptext und Gerätetext die folgenden Richtlinien:

- Wenn Sie eingebettete Schriftarten für dynamische oder Eingabetextfelder verwenden, sollten Sie nur Konturen für die Zeichen einbetten, die auch tatsächlich angezeigt werden müssen. Dies trägt zu einer geringeren Dateigröße bei. Wenn beispielsweise das Alter des Benutzers als Zahl in ein Eingabetextfeld eingegeben werden soll, betten Sie nur die Konturen für die Ziffern 0–9 ein. In diesem Fall sollten Sie die Zeicheneingabe auf Zahlen beschränken.
- Der Adobe Device Central-Emulator emuliert keine Geräteschriftarten, es sei denn, dieselben Schriftarten sind auch auf dem Computer installiert, auf dem Sie den Inhalt entwickeln. Deshalb kann das Textfeld mit der Geräteschriftart im Emulator anders aussehen und angeordnet sein als auf dem Gerät.
- Wenn Sie eine generische Geräteschriftart anwenden (_sans, _serif oder _typewriter), sucht Flash Lite auf dem Gerät eine ähnliche Schriftart für die Anzeige des Textes. Da Mobilgeräte in der Regel jedoch über weniger Schriftarten und Schriftstile verfügen als Desktopcomputer, ist es eventuell nicht möglich, eine Schriftart wie _sans einer serifenlosen Schriftart zuzuordnen. Sie müssen die Anwendung auf jedem Zielgerät testen, um die richtigen Einstellungen für die Schriftart zu ermitteln.

Anti-Alias-Text in Flash Lite ist im Prinzip eine komplexe Vektorform, deren Darstellung wie jede andere Vektorform relativ viel Prozessorleistung erfordert. Da die Verarbeitungsgeschwindigkeit der meisten Geräte eher niedrig ist, kann die Animation von sehr viel Anti-Alias-Text zu Leistungseinbußen der Anwendung führen. Um die Leistung zu verbessern, empfiehlt es sich möglicherweise, die Wiedergabequalität des Flash Lite-Players während der Animation vorübergehend zu verringern und nach Abschluss der Animation wieder die höhere Wiedergabequalität zu verwenden.

Für ein Textfeld eine Methode zur Schriftwiedergabe auswählen

1 Wählen Sie auf der Bühne ein Textfeld aus.

2 Wählen Sie im Eigenschafteninspektor im Popupmenü „Methode zur Schriftwiedergabe“ eine der folgenden Optionen aus:



- Wählen Sie „Geräteschriftarten verwenden“, wenn Flash Lite eine auf dem Gerät verfügbare Schriftart verwenden soll. In die veröffentlichte SWF-Datei werden keine Schriftartdaten eingebettet.
- Wählen Sie „Bitmaptext (kein Anti-Alias)“, wenn Flash Lite Schriftkonturen an Pixelbegrenzungen ausrichten soll, damit kleingedruckter Text klar und deutlich zu erkennen ist. Diese Option setzt voraus, dass Sie die Schriftkonturen in die veröffentlichte SWF-Datei einbetten. „Schriftkonturen in SWF-Dateien einbetten“ auf Seite 66
- Wählen Sie „Anti-Aliasing für Animation“, wenn Flash Lite die Schriftart des Textfelds anhand der aktuellen Einstellung für die Wiedergabequalität glätten soll (siehe („Wiedergabequalität und Anti-Alias-Text in Flash Lite“ auf Seite 65). Diese Option setzt voraus, dass Sie die Schriftkonturen in die veröffentlichte SWF-Datei einbetten.

Wiedergabequalität und Anti-Alias-Text in Flash Lite

Verwenden Sie die Eigenschaft `MovieClip._quality`, um zu steuern, wie Vektor- und Bitmapgrafiken in Flash Lite wiedergegeben werden. Mögliche Werte sind „`LOW"`, „`MEDIUM"`, „`HIGH"` und „`BEST"`. Der Standardwert ist„`HIGH"`. Diese Eigenschaft kann für jedes Movieclip-Objekt festgelegt werden. Um diese Eigenschaft global festzulegen, müssen Sie den Code `_quality = "lowMediumHighOrBest"` in Bild 1 platzieren.

Flash Lite gibt Anti-Alias-Text mithilfe von Vektordarstellungen der Schriftkonturen wieder. Wenn Anti-Alias-Text möglichst glatt aussehen soll, sollten Sie die hohe Einstellung für die Wiedergabequalität beibehalten. In der folgenden Abbildung wird gezeigt, wie ein Textfeld mit Anti-Alias-Text (Arial, 24 Punkt) bei einer hohen, einer mittleren und einer niedrigen Wiedergabequalität aussieht:

High
quality
setting

Zur Maximierung der Animationsleistung und der Bildrate (z. B. während einer intensiven Animations- oder Tweening-Sequenz) können Sie die Wiedergabequalität vorübergehend auf eine niedrigere Einstellung zurücksetzen und nach dem Ende der Animation die ursprüngliche Einstellung reaktivieren.

Verwenden Sie den Befehl `fscommand2("SetInputTextType")`, um die Eingabetextqualität festzulegen. Weitere Informationen hierzu finden Sie im Eintrag im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.

Schriftkonturen in SWF-Dateien einbetten

Zur Wiedergabe der Schriftart eines Textfelds kann Flash Lite entweder die auf dem Gerät verfügbaren Schriftarten verwenden oder aber Schriftkonturen, die in die veröffentlichte SWF-Datei eingebettet wurden (siehe „Methoden zur Schriftwiedergabe in Flash Lite“ auf Seite 64). Durch das Einbetten von Schriftkonturen in die SWF-Datei lässt sich gewährleisten, dass die Schriftart des Textfelds auf allen Zielplattformen gleich aussieht. Allerdings wird dadurch auch die Dateigröße erhöht. Zur Wiedergabe von Bitmaptext (kein Anti-Alias) oder Anti-Alias-Text benötigt Flash Lite Schriftkonturen.

Bei statischen Textfeldern, denen die Wiedergabemethode „Anti-Aliasing für Animation“ oder „Bitmaptext (kein Anti-Alias)“ zugewiesen wurde, bettet Flash die zur Anzeige der Textfeldinhalte erforderlichen Schriftkonturen automatisch ein. Wenn ein statisches Textfeld beispielsweise das Wort *Fertig* enthält, werden die Schriftkonturen, die zur Anzeige der sechs Zeichen benötigt werden, automatisch eingebettet (also *F*, *e*, *r*, *t*, *i* und *g*). Da sich der Inhalt eines statischen Textfelds nicht ändern kann, müssen nur die Schriftkonturen dieser Zeichen berücksichtigt werden.

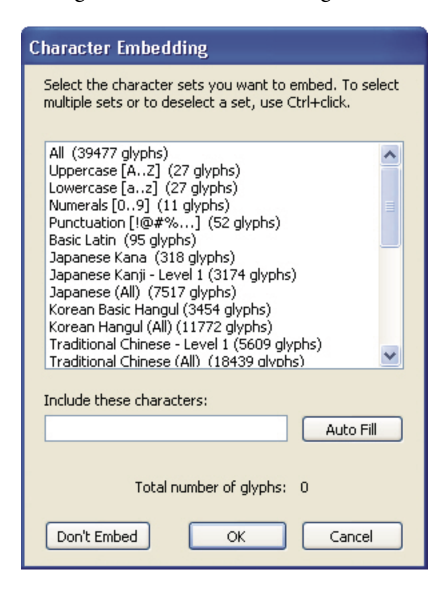
Bei dynamischen Textfeldern und Eingabetextfeldern, denen die Wiedergabemethode „Anti-Aliasing für Animation“ oder „Bitmaptext (kein Anti-Alias)“ zugewiesen wurde, müssen Sie die Zeichen angeben, deren Schriftkonturen in die veröffentlichte SWF-Datei eingebettet werden sollen. Da sich der Inhalt dieser Textfelder während der Wiedergabe ändern kann, kann Flash nicht vorherbestimmen, welche Schriftkonturen benötigt werden. Sie können Schriftkonturen für alle Zeichen einer ausgewählten Schriftart, für einen bestimmten Zeichenbereich oder für bestimmte Zeichen einbetten. Im Dialogfeld „Zeicheneinbettung“ können Sie angeben, welche Zeichen in die veröffentlichte SWF-Datei eingebettet werden sollen.

Schriftkonturen für dynamische Textfelder oder Eingabetextfelder einbetten

1 Wählen Sie das dynamische Textfeld oder das Eingabetextfeld auf der Bühne aus.

2 Wählen Sie im Eigenschafteninspektor im Popupmenü „Methode zur Schriftwiedergabe“ die Option „Bitmaptext (kein Anti-Alias)“ oder „Anti-Aliasing für Animation“ aus.

3 Klicken Sie neben dem Popupmenü „Methode zur Schriftwiedergabe“ auf die Schaltfläche „Einbetten“, um das Dialogfeld „Zeicheneinbettung“ zu öffnen.



4 Sie können entweder die einzubettenden Zeichen in der Liste auswählen, spezifische Zeichen in das Textfeld „Diese Zeichen einschließen“ eingeben oder auf „Auto-Ausfüllen“ klicken, um nur die im ausgewählten Textfeld vorhandenen Zeichen zu berücksichtigen.

5 Klicken Sie auf „OK“.

Text mit Bildlauf

Text mit Bildlauffunktion erstellen

Flash Lite unterstützt die Eigenschaften `TextField.scroll` und `TextField.maxscroll`, mit denen Sie Textfelder mit Bildlauffunktion erstellen können. Die Eigenschaft `scroll` gibt die erste sichtbare Zeile eines Textblocks an; Sie können den zugehörigen Wert abrufen und festlegen. Mit dem folgenden Code wird z. B. für das Textfeld mit dem Variablennamen `story_text` ein Bildlauf um fünf Zeilen nach unten durchgeführt:

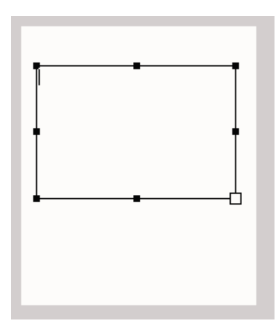
```
story_text.scroll += 5;
```

Die Eigenschaft `maxscroll` gibt die erste sichtbare Zeile in einem Textblock an, wenn die letzte Textzeile in diesem Block sichtbar ist; diese Eigenschaft ist schreibgeschützt. Durch einen Vergleich der Eigenschaften `maxscroll` und `scroll` eines Textfelds lässt sich feststellen, wie weit ein Benutzer für ein Textfeld bereits einen Bildlauf durchgeführt hat. Dies ist nützlich, wenn Sie eine Bildlaufleiste erstellen möchten, die Aufschluss über die aktuelle Bildlaufposition des Benutzers im Verhältnis zur maximalen Bildlaufposition gibt.

Textfeld mit Bildlauffunktion erstellen und über ActionScript steuern

1 Erstellen Sie in Flash ein neues Mobilgerätdokument.

2 Erstellen Sie mit dem Textwerkzeug auf der Bühne ein Textfeld, dessen Größe ungefähr dem Textfeld in der folgenden Abbildung entspricht:



3 Wählen Sie im Eigenschafteninspektor im Popupmenü „Zeilentyp“ die Option „Mehrzeilig“ aus.

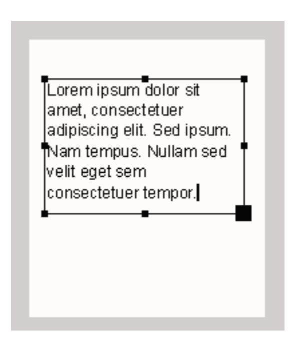
4 Wählen Sie im Eigenschafteninspektor im Popupmenü „Texttyp“ die Option „Dynamischer Text“ aus.

5 Wählen Sie im Eigenschafteninspektor im Popupmenü „Methode zur Schriftwiedergabe“ die Option „Geräteschriftarten verwenden“ aus.

6 Wählen Sie Text > Rollbar, damit ein Bildlauf für das Textfeld durchgeführt werden kann.

7 Geben Sie im Eigenschafteninspektor im Textfeld „Instanzname“ den Text **story** ein.

8 Doppelklicken Sie im Textfeld und geben Sie mindestens eine Zeile mehr Text ein als in das Textfeld passt.



9 Wählen Sie in der Zeitleiste das erste Bild in Ebene 1 aus und öffnen Sie das Bedienfeld „Aktionen“.

10 („Fenster“ > „Aktionen“). Geben Sie im Bedienfeld „Aktionen“ den folgenden -Code ein:

```
var keyListener:Object = new Object();
keyListener.onKeyDown = function() {
    var keyCode = Key.getCode();
    if (keyCode == Key.DOWN) {
        story.scroll++;
    }
    else if (keyCode == Key.UP) {
        story.scroll--;
    }
};
Key.addListener(keyListener);
```

11 Wählen Sie Steuerung > „Film testen“, um die Anwendung im Adobe Device Central-Emulator zu testen.

Klicken Sie im Emulator auf die Navigationstasten <Nach-oben> und <Nach-unten> (oder drücken Sie die Pfeiltasten <Nach-oben> und <Nach-unten> auf der Computertastatur), um einen Bildlauf nach oben oder nach unten durchzuführen.

Kapitel 6: Mit Sounds arbeiten

Adobe Flash Lite unterstützt zwei Arten von Sounds: standardmäßigen (*nativen*) Flash-Sound und Gerätesound. Native Sounds werden direkt vom Flash Lite Player wiedergegeben, genau wie in der Desktopversion von Adobe Flash Player. Gerätesounds werden dagegen direkt vom Gerät wiedergegeben, nicht vom Flash Lite Player. Zu den Formaten für Gerätesounds zählen beispielsweise MIDI und MFi, die jeweils unterstützten Soundformate richten sich jedoch nach dem Gerät.

Gerätesound

Gerätesound

Gerätesounds werden direkt vom Gerät wiedergegeben, nicht vom Flash Lite Player. Verschiedene Geräte können unterschiedliche Soundformate unterstützen, wie MIDI, MFi oder MP3. Zur Verwendung von Gerätesounds in Flash Lite-Inhalten können Sie sie entweder in die veröffentlichte SWF-Datei einbetten oder Sie können externe Sounddateien von einem Speicherort im Netzwerk oder vom lokalen Dateisystem des Geräts laden. Abgesehen von einigen Ausnahmen können Sie Gerätesounds mit dem Sound-Objekt von ActionScript genauso steuern wie Sounds in der Desktopversion von Flash Player.

Gerätesound-Bundle verwenden

Um einen Gerätesound als Bundle in Ihre Anwendung einzubetten, müssen Sie zunächst einen Proxy-Sound in einem Format importieren, das vom Flash-Authoring-Tool erkannt wird, wie beispielsweise eine MP3-, WAV- oder AIFF-Datei. Dann verknüpfen Sie den Proxy-Sound mit einer Gerätesounddatei auf Ihrem Computer, die Sie als Bundle in Ihre Anwendung integrieren möchten. Während der Veröffentlichung der SWF-Datei ersetzt das Flash-Authoring-Tool den Proxy-Sound durch den verknüpften externen Sound. Bei der Wiedergabe gibt Flash Lite die Sounddaten zur Dekodierung und Wiedergabe an das Gerät zurück.

Sie können auch mehrere Gerätesounds in verschiedenen Formaten in einem Flash-Sound-Bundle (FLS-Datei) zusammenfassen. Dies ist nützlich, wenn ein und derselbe Inhalt auf mehreren Geräten wiedergegeben werden soll, die unterschiedliche Gerätesoundformate unterstützen. Weitere Informationen finden Sie unter „Sound-Bundle erstellen“ auf Seite 73.

Gerätesound importieren und wiedergeben

Zur Wiedergabe eines Gerätesounds können Sie den Sound der Zeitleiste zuweisen oder das Sound-Objekt von ActionScript verwenden.

1 Erstellen Sie in Flash ein neues Mobilgerätdokument, und speichern Sie es unter **device_sound.fla**.

2 Wählen Sie „Datei“ > „Importieren“ > „In Bibliothek importieren“. Suchen Sie auf der Seite mit Beispielen und Tutorials unter www.adobe.com/go/learn_flt_samples_and_tutorials_de nach der komprimierten Datei für Ihre Flash Lite-Version, laden Sie sie herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner „Samples“.

3 Wählen Sie die Datei proxy.wav aus und klicken Sie auf „OK“.

Hinweis: *Als Proxy-Sound kann jede Sounddatei verwendet werden, die vom Flash-Authoring-Tool erkannt wird. Die vordefinierte Datei proxy.wav soll Ihnen in diesem Beispiel die Arbeit erleichtern.*

1 So verknüpfen Sie den Proxy-Sound mit der Gerätesounddatei:

a Klicken Sie im Bedienfeld „Bibliothek“ mit der rechten Maustaste (Windows) bzw. bei gedrückter Taste <Ctrl> (Macintosh) auf das Symbol des Proxy-Sounds und wählen Sie im Kontextmenü die Option „Eigenschaften“ aus, um das Dialogfeld „Soundeigenschaften“ zu öffnen.

b Klicken Sie im Dialogfeld „Soundeigenschaften“ rechts neben dem Textfeld „Gerätesound“ auf das Ordnersymbol, um das Dialogfeld zur Auswahl eines Gerätesounds zu öffnen.

c Wechseln Sie zu www.adobe.com/go/learn_flt_samples_and_tutorials_de. Suchen Sie auf der Seite mit Beispielen und Tutorials nach der komprimierten Datei (ZIP) für Ihre Flash Lite-Version, laden Sie sie herunter, dekomprimieren Sie sie und wechseln Sie in den Ordner Samples und wählen Sie die hi_hat.mid genannte Datei aus.



d (Optional) Um den Gerätesound mithilfe von ActionScript zu steuern, klicken Sie auf „Erweitert". Wählen Sie dann in den erweiterten Soundeigenschaften die Option „Export für ActionScript" aus und geben Sie **device_sound** in das Textfeld „Bezeichner" ein.



e Klicken Sie auf „OK", um das Dialogfeld „Soundeigenschaften" zu schließen.

Zur Wiedergabe des Gerätesounds können Sie den Proxy-Sound der Zeitleiste zuweisen oder das Sound-Objekt von ActionScript verwenden. Wenn Sie das Sound-Objekt von ActionScript verwenden möchten, fahren Sie mit Schritt 6 fort.

2 So weisen Sie den Gerätesound der Zeitleiste zu:

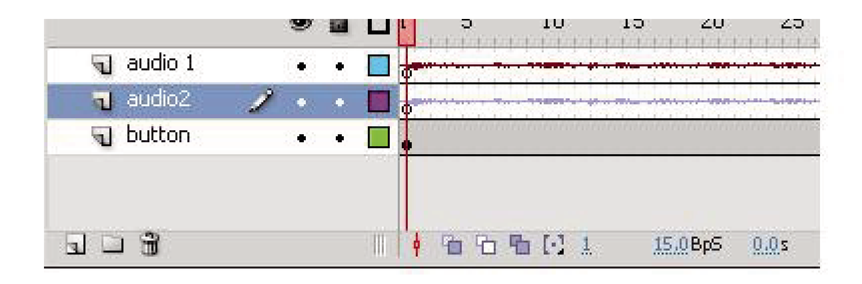
a Wählen Sie das Schlüsselbild bei Bild 1 in der Ebene „Content" aus.

b Wählen Sie im Eigenschafteninspektor im Popupmenü „Sound" die Datei proxy.wav aus.

Dadurch wird der Proxy-Sound mit dem Schlüsselbild verbunden.



Das Flash-Authoring-Tool zeigt die Wellenform des Proxy-Sounds in der Zeitleiste an. Sounds, die mit externen Gerätesounds verknüpft sind, haben grüne Wellenformen, die anderen Sounds blaue Wellenformen, wie in der folgenden Abbildung gezeigt.



A. Mit einem externen Gerätesound verknüpfter Sound B. Mit keinem externen Gerätesound verknüpfter Sound

3 So geben Sie den Sound mit ActionScript wieder:

4 Wählen Sie in der Zeitleiste die Ebene „Actions“ aus.

5 Öffnen Sie das Bedienfeld „Aktionen“ („Fenster“ > „Aktionen“) und geben Sie folgenden Code ein:

```
var deviceSound:Sound = new Sound();
deviceSound.attachSound("device_sound");
deviceSound.start();
```

6 Wählen Sie „Steuerung“ > „Film testen“ , um den Adobe Device Central-Emulator zu starten und die SWF-Datei zu testen.

Sound-Bundle erstellen

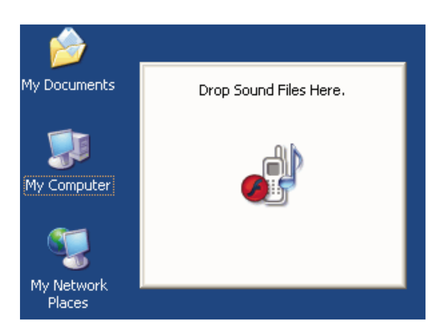
In Flash Lite können Sie mehrere Gerätesounds in verschiedenen Formaten in einem *Sound-Bundle* zusammenfassen. So könnte eine einzelne Flash-Anwendung ein und denselben Sound sowohl im MIDI- als auch im MFi-Format enthalten. Wenn die Anwendung auf einem Gerät ausgeführt wird, das MIDI-Audio unterstützt, wählt Flash Lite die MIDI-Sounddaten im Sound-Bundle aus und übergibt sie zur Wiedergabe an das Gerät. Wird die Anwendung auf einem Gerät ausgeführt, das nur MFi unterstützt, übergibt Flash Lite die MFi-Sounddaten an das Gerät.

Mit dem Dienstprogramm Flash Lite Sound Bundler können Sie eine Sound-Bundle-Datei (FLS) erstellen. Dann verknüpfen Sie die FLS-Datei mit einem Proxy-Sound im Flash Lite-Dokument, genau wie bei einem einzelnen Gerätesound. Weitere Informationen zum Einfügen von Gerätesounds in Flash Lite-Anwendungen finden Sie unter „Gerätesound-Bundle verwenden“ auf Seite 69.

***Hinweis:** Das Sound Bundler-Dienstprogramm wird derzeit nur von Windows-Systemen unterstützt.*

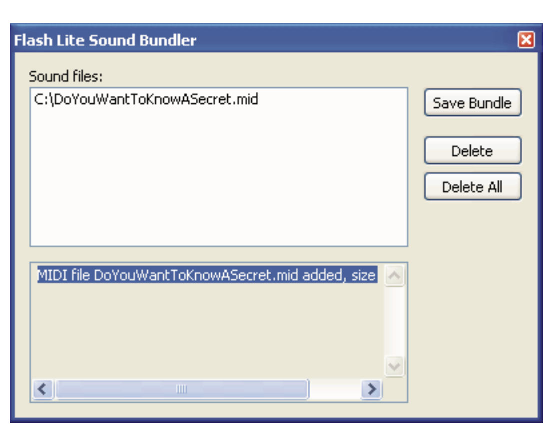
1 Sie können die Flash Lite Sound Bundler Anwendung (FlashLiteBundler.exe) unter www.adobe.com/go/developer_flashlite_de finden und herunterladen.

2 Sound Bundler wird als nicht verankertes Fenster angezeigt.



3 Ziehen Sie die erste Sounddatei, die Sie mit anderen Sounds in einem Bundle zusammenfassen möchten, vom Desktop in das Sound Bundler-Fenster.

Daraufhin wird ein neues Fenster eingeblendet, in dem der Inhalt des Sound-Bundles aufgeführt wird. Im unteren Bereich des Fensters werden Informationen zu den Sounds im Sound-Bundle angezeigt, z. B. Details zu den Soundformaten, zur Größe der Sounddaten und die Dateinamen.



***A.** Informationen zu Dateien im Sound-Bundle **B.** Liste der Dateien im Sound-Bundle*

4 Ziehen Sie die übrigen Sounddateien, die Sie dem Bundle hinzufügen möchten, in das Fenster.

Einem Bundle kann pro Audioformat nur jeweils eine Datei hinzugefügt werden, d. h. eine FLS-Datei kann z. B. nicht zwei MIDI-Dateien enthalten.

5 Um eine Datei aus dem Sound-Bundle zu löschen, wählen Sie sie in der Liste aus und klicken Sie auf „Löschen“.

Um alle Dateien aus dem Sound-Bundle zu löschen, klicken Sie auf die Schaltfläche „Alle löschen“.

6 Zum Speichern des Sound-Bundles klicken Sie auf „Zusammenfassung speichern“ und wählen Sie einen Namen und einen Speicherort für die FLS-Datei aus.

7 Um den Sound Bundler zu beenden, klicken Sie in seinem Fenster auf die Schaltfläche zum Schließen (X).

Als Nächstes müssen Sie die Sound-Bundle-Datei (FLS) Ihrem Flash-Dokument hinzufügen. Dabei verfahren Sie ähnlich wie beim Einfügen eines normalen Gerätesounds in ein Flash-Dokument. Anstatt aber eine einzelne Gerätesounddatei anzugeben, die den Proxy-Sound ersetzen soll, müssen Sie die von Ihnen erstellte FLS-Datei angeben. Weitere Informationen finden Sie unter „Gerätesound“ auf Seite 69.

Externe Gerätesounds wiedergeben

Sie können nicht nur Gerätesounds wiedergeben, die in der veröffentlichten SWF-Datei als Bundle enthalten sind (siehe „Gerätesound-Bundle verwenden“ auf Seite 69), sondern auch externe Sounddateien laden und wiedergeben. Zum Wiedergeben von externen Gerätesounds verwenden Sie die Methode `loadSound()` des Sound-Objekts. Wie bei Gerätesound-Bundles übergibt der Flash Lite Player die geladenen externen Audiodaten zur Dekodierung und Wiedergabe an das Gerät.

Im Folgenden finden Sie wichtige Informationen zur Wiedergabe von externen Gerätesounds in Flash Lite:

- Im Unterschied zur Desktopversion von Flash Player werden geladene externe Sounds in Flash Lite immer als Ereignissounds interpretiert. Dies bedeutet, dass externe Gerätesounds nicht im Streaming-Verfahren direkt beim Download wiedergegeben werden. Die Sounddaten müssen vollständig heruntergeladen werden, bevor sie wiedergegeben werden können. Deshalb müssen Sie auch die Methode `start()` des Sound-Objekts aufrufen, um den Sound nach dem Laden wiederzugeben (siehe folgendes Codebeispiel).
- In Flash Lite wird der zweite Parameter der Methode `loadSound()` nicht unterstützt (*isStreaming*). Falls dieser Parameter vorhanden ist, wird er von Flash Lite ignoriert.
- In Flash Lite werden geladene externe MP3-Dateien nicht nativ wiedergegeben. Wenn Ihre Anwendung eine externe MP3-Datei lädt, gibt Flash Lite die Sounddaten zur Dekodierung und Wiedergabe an das Gerät weiter, wie bei jeder anderen geladenen externen Sounddatei.

Im folgenden Code wird gezeigt, wie eine externe Sounddatei geladen und wiedergegeben wird:

```
// Create the sound object.
var mySound:Sound = new Sound();
// Define onLoad handler for sound,
// which starts the sound once it has fully loaded.
mySound.onLoad = function(success){
    if(success == true) {
        mySound.start();
    }
}
// Load the sound.
mySound.loadSound("http://www.adobe.com/audio.midi");
```

Gerätesounds mit Animationen synchronisieren

Gerätesounds werden in Flash Lite grundsätzlich wie Ereignissounds verarbeitet. Deshalb können Gerätesounds nicht so mit Animationen in der Zeitleiste synchronisiert werden, wie dies bei nativen Flash-Sounds der Fall ist. Sie können die Synchronisierung von Gerätesounds jedoch auf ungefähre Weise simulieren, indem Sie die Eigenschaft `_forceframerate` des Flash Lite Players auf `true` einstellen. Wenn diese Eigenschaft auf `true` eingestellt ist, überspringt Flash Lite Bilder in der Animation, um die festgelegte Bildrate der SWF-Datei beizubehalten. Wenn die Gerätesounddaten beim Authoring auf die richtige Dauer eingestellt und vom Gerät mit der erwarteten Bildrate wiedergegeben werden, kann so eine weitgehende Synchronisation von Animation und Sound erreicht werden.

Angenommen, ein Gerätesound mit einer Länge von fünf Sekunden soll während der Wiedergabe mit einer Animation in der Zeitleiste synchronisiert werden. Die Bildrate der Anwendung ist auf 15 fps eingestellt Wenn Sie den Sound starten, indem Sie ihn einem Bild in der Zeitleiste zuweisen oder `Sound.start()` aufrufen, stellen Sie die Eigenschaft `_forceframerate = "true"` gleichzeitig ein. Anschließend stellt Flash Lite bei der Wiedergabe des Gerätesounds sicher, dass der Abspielkopf sich in jeder Sekunde um 15 Bilder in der Zeitleiste nach vorne bewegt. Wenn der Player in diesem Zeitraum nicht jedes Bild in der Animation wiedergeben kann, aus welchem Grund auch immer, werden Bilder übersprungen, um die angegebene Bildrate beizubehalten.

Weitere Informationen zur `_forceframerate`-Eigenschaft finden Sie unter `_forceframerate-Eigenschaft` im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.

Unterstützte Gerätesoundformate ermitteln

Die Eigenschaft `System.capabilities.audioMIMETypes` enthält ein Array der MIME-Audiotypen, die vom Gerät unterstützt werden. Anhand dieser Informationen können Sie ermitteln, welche Gerätesoundtypen Ihre Anwendung wiedergeben kann. Die Array-Indizes entsprechen den unterstützten MIME-Typen, sodass Sie schnell und einfach feststellen können, ob ein Gerät einen bestimmten Audiotyp unterstützt.

Der folgende Beispielcode ermittelt zunächst, ob das Gerät die Wiedergabe von MIDI-Audio unterstützt, bevor eine externe MIDI-Datei geladen wird:

```
if (System.capabilities.audioMIMETypes["audio/midi"]) {
    my_sound.loadSound("soundtrack.mid");
}
```

Nativer Sound

Native Flash-Sounds

Zusätzlich zu Gerätesounds unterstützt Flash Lite auch standardmäßige *native* Flash-Sounds. Ein nativer Sound ist im Prinzip ein Sound in einem Format, das vom Flash-Authoring-Tool erkannt wird. In Flash Lite können native Sounds entweder als Ereignissounds oder als synchronisierte Sounds wiedergegeben werden (siehe „Gerätesound“ auf Seite 69).

Das Verfahren, mit dem native Sounds in Flash Lite-Anwendungen verwendet werden, stimmt mit dem in Flash-Desktop-Anwendungen erforderlichen Verfahren weitgehend überein. Dabei gelten für Flash Lite jedoch folgende Besonderheiten:

- Flash Lite bietet keine Unterstützung für die *native* Wiedergabe von externen MP3-Dateien. Sie können jedoch externe MP3-Dateien in Ihre Anwendung laden, wenn das Gerät MP3-Audio wiedergeben kann. Weitere Informationen finden Sie unter „Externe Gerätesounds wiedergeben“ auf Seite 75.
- Flash Lite bietet keine Unterstützung für die Audiokomprimierungsoption „Sprache“.

Weitere Informationen zur Verwendung von nativen Sounds in Flash finden Sie im Handbuch *Flash verwenden* in den folgenden Themen von „Mit Sounds arbeiten“.

8-kHz-Abtastrate verwenden

Standardmäßig exportiert das Flash-Authoring-Tool natives Audio mit einer Abtastrate von 5, 11, 22 oder 44 kHz (Kilohertz). Die Wiedergabe von Audiodaten mit diesen Abtastraten wird jedoch von zahlreichen Geräten nicht unterstützt. Beispielsweise unterstützen Geräte der Nokia Serie 60 nur 8 und 16 kHz. In bisherigen Versionen von Flash Lite wurden die Audiodaten zur Laufzeit mit einer Abtastrate von 8 kHz neu erstellt, bevor die nativen Sound-APIs des Geräts zur Wiedergabe des Sounds aufgerufen wurden. Dieses so genannte Resampling erfordert jedoch zusätzliche Arbeitsspeicher- und Prozessorressourcen.

In Adobe Flash CS4 Professional können Sie festlegen, dass das Authoring-Tool während der Veröffentlichung der SWF-Datei die nativen Audiodaten der Anwendung mit einer Abtastrate von 8 kHz neu erstellt. Sie können diese Einstellung auf alle Streaming- und Ereignissounds der Anwendung oder nur auf bestimmte Sounds im Bedienfeld „Bibliothek“ des Dokuments anwenden.

Im Folgenden wird zunächst beschrieben, wie die 8-kHz-Abtastrate im Dialogfeld „Soundeigenschaften“ für einzelne Sounds aktiviert wird. Anschließend erfahren Sie, wie Sie diese Option im Dialogfeld „Einstellungen für Veröffentlichungen“ global für alle Streaming- und Ereignissounds anwenden können.

Hinweis: *Für Streaming-Sounds verwendet Flash immer die globalen Optionen für die Soundkomprimierung, die im Dialogfeld „Einstellungen für Veröffentlichungen“ festgelegt sind. Die Komprimierungsoptionen, die Sie im Dialogfeld „Soundeigenschaften“ für spezifische Sounds festlegen, gelten nur für Ereignissounds.*

8-kHz-Abtastrate für einen einzelnen Sound aktivieren

1 Klicken Sie in Flash mit der rechten Maustaste (Windows) bzw. bei gedrückter Ctrl-Taste (Macintosh) auf ein Sound-Symbol im Bedienfeld „Bibliothek“ und wählen Sie im Kontextmenü die Option „Eigenschaften“ aus.

2 Wählen Sie im Dialogfeld „Soundeigenschaften“ im Popupmenü „Komprimierung“ die Option „MP3“ aus.

   Die 8-kHz-Abtastrate ist nur für Audio verfügbar, das im MP3-Format komprimiert ist.

3 Deaktivieren Sie die Option „Qualitätseinstellung des importierten MP3 verwenden“, falls sie ausgewählt ist.

4 Aktivieren Sie die Option „8-kHz-Abtastrate verwenden“.



A. 8-kHz-Abtastrate verwenden

5 Klicken Sie auf „OK“.

8-kHz-Abtastrate global für alle nativen Sounds aktivieren

1 Wählen Sie in Flash „Datei“ > „Einstellungen für Veröffentlichungen“.

2 Klicken Sie im Dialogfeld „Einstellungen für Veröffentlichungen“ auf die Registerkarte „Flash“.

3 Führen Sie folgende Schritte aus, um die 8-kHz-Abtastrate für alle Streaming-Sounds in Ihrer Anwendung zu aktivieren:

a Klicken Sie auf „Optionen für Streaming-Sounds einstellen".

b Wählen Sie im Dialogfeld „Soundeigenschaften" im Popupmenü „Komprimierung" die Option „MP3" aus.

Die 8-kHz-Abtastrate ist nur für Audio verfügbar, das im MP3-Format komprimiert ist.

c Aktivieren Sie die Option „8-kHz-Abtastrate verwenden".



d Klicken Sie auf „OK", um das Dialogfeld „Einstellungen für Veröffentlichungen" zu schließen.

4 Führen Sie folgende Schritte aus, um die 8-kHz-Abtastrate für alle Ereignissounds zu aktivieren:

a Klicken Sie auf „Optionen für Ereignissounds einstellen".

b Wählen Sie im Dialogfeld „Soundeigenschaften" im Popupmenü „Komprimierung" die Option „MP3" aus.

Hinweis: *Die 8-kHz-Abtastrate ist nur für Audio verfügbar, das im MP3-Format komprimiert ist.*

c Aktivieren Sie die Option „8-kHz-Abtastrate verwenden".

d Klicken Sie auf „OK".

5 Klicken Sie auf „OK", um das Dialogfeld „Einstellungen für Veröffentlichungen" zu schließen.

Kapitel 7: Mit Videos arbeiten

Video kann sowohl in Flash Lite 2.x- als auch in Flash Lite 3.x-Anwendungen hinzugefügt werden. In Flash Lite 2.x-Anwendungen wird das Video durch das Mobilgerät dekodiert und wiedergegeben. Dieser Videotyp wird als *Gerätevideo* bezeichnet. Flash Lite 3.x-Anwendungen unterstützen zwei Arten von Video: Gerätevideo und FLV-Dateien (Flash Video). Wenn Sie eine FLV-Datei in einer Anwendung abspielen, wird das Video mit Flash Lite 3.x dekodiert und wiedergegeben.

Mit FLV-Dateien arbeiten

Informationen zur FLV-Dateiunterstützung in Flash Lite 3.0 und höher

Flash Lite 3.0 unterstützt nun FLV-Dateien (Flash Video), in denen Versionen der für Mobilgeräte optimierten On2- und Sorenson-Codecs verwendet werden. Die FLV-Dateien werden mit Flash Lite 3.x und nicht durch das Gerät wiedergegeben. Falls ein Gerät Flash Lite 3.x unterstützt, können Sie Ihrer Anwendung FLV-Dateien hinzufügen.

Hinweis: *Einige Gerätehersteller unterstützen unter Umständen keine FLV-Dateien, falls bereits ein ausreichender Mechanismus zur Wiedergabe von Gerätevideo vorhanden ist. Verwenden Sie in diesem Fall das Gerätevideo in den Anwendungen. Überprüfen Sie auf der Registerkarte „Geräteprofile“ in Adobe Device Central, ob das Gerät FLV-Dateien (Flash Video) unterstützt.*

FLV-Dateien können folgendermaßen in eine Anwendung integriert werden:

- Betten Sie die FLV-Dateien in die Anwendung ein.
- Geben Sie externe FLV-Dateien von einer lokalen Festplatte oder von einem Webserver aus über HTTP wieder.
- Geben Sie externe FLV-Dateien vom Flash Media Server aus über RTMP wieder.

Gehen Sie folgendermaßen vor, um Video (wie QuickTime- oder Windows Media-Videodateien) in FLV-Dateien zu konvertieren:

- Verwenden Sie einen Encoder (wie Adobe Flash Video Encoder oder Sorenson™ Squeeze).
- Importieren Sie die Datei in das Flash-Authoring-Tool und exportieren Sie sie anschließend als FLV-Datei.
- Verwenden Sie das Plugin FLV Export, um FLV-Dateien von unterstützten Video-Bearbeitungsprogrammen aus zu exportieren.

Folgende Videofunktionen stehen in Flash Lite 3.x nicht zur Verfügung:

- Die Camera-Klasse oder das Aufzeichnen von Video.
- RTMPT- und RTMPS-Verbindungen. Diese Verbindungsversuche werden automatisch zu RTMP-Verbindungen.
- Die bidirektionale Kommunikation zwischen Clients. Gemeinsame Remote-Objekte werden nicht unterstützt. Clients können keine freigegebenen Objekte gemeinsam nutzen.
- Alpha-Kanal-Video.
- Die FLVPlayback-Komponente. Weitere Informationen zu Komponenten finden Sie unter „Komponenten in Flash Lite“ auf Seite 1.

Siehe auch

„FLV-Dateien in eine Anwendung einbetten“ auf Seite 80

„Externe FLV-Dateien in einer Anwendung wiedergeben“ auf Seite 81

FLV-Dateien in eine Anwendung einbetten

Importieren Sie die FLV-Datei in die Dokumentbibliothek, um sie einzubetten. Bei der Veröffentlichung der Anwendung bettet das Authoring-Tool die FLV-Datei in die veröffentlichte SWF-Datei ein.

Die eingebetteten FLV-Dateien werden an die Zeitleiste angehängt. Wenn beispielsweise ein Flash-Dokument 10 Bilder pro Sekunde wiedergibt, füllt eine FLV-Datei mit einer Länge von 10 Sekunden die Zeitleiste von Bild 1 bis 100 aus. Verwenden Sie Adobe ActionScript, um die Wiedergabe der eingebetteten FLV-Datei zu steuern. Sie können die Datei abspielen, anhalten oder den Abspielkopf in der Zeitleiste verschieben. Außerdem können Sie dem eingebetteten Video-Objekt einen Instanznamen geben und es über die Eigenschaften des Video-Objekts in ActionScript steuern.

Durch Einbetten der FLV-Dateien können die Dateien mit anderen Elementen in dem Dokument synchronisiert werden. So können Sie beispielsweise interaktive Elemente für bestimmte Bilder hinzufügen, die mit anderen Dokumenten verknüpft sind. Das Einbetten von FLV-Dateien empfiehlt sich insbesondere für kurze Dateien ohne Audiospur. Durch das Einbetten von FLV-Dateien kann sich die Größe der veröffentlichten SWF-Datei erheblich vergrößern.

Hinweis: *In Adobe Flash Lite 2.0-Anwendungen können Sie Gerätevideo in eine Anwendung einbetten. Dieses Verfahren wird als „Videobündelung“ bezeichnet. Siehe auch „Gerätevideo in eine Anwendung importieren“ auf Seite 84.*

1 Wählen Sie in Flash „Datei“ > „Importieren“ > „Video importieren“.

   Der Videoimportassistent wird eingeblendet.

2 Klicken Sie auf die Option zum Importieren einer Datei, die auf Ihrem Computer gespeichert ist. Klicken Sie dann auf „Durchsuchen“.

3 Wechseln Sie zum Ordner mit der FLV-Datei und wählen Sie sie aus.

   Falls die Videodatei nicht in der Liste aufgeführt ist (oder nicht ausgewählt werden kann), wählen Sie „Alle Dateien“ (*.*) aus dem Popup-Menü mit den verfügbaren Dateitypen (Windows) oder wählen Sie „Alle Dateien“ aus dem Popup-Menü „Aktivieren“ (Macintosh).

4 Klicken Sie auf „Öffnen“. Klicken Sie im Videoimportassistenten auf „Weiter“.

5 Wählen Sie im Bereitstellungsbildschirm „Video in SWF einbetten und in Zeitleiste wiedergeben“ und klicken Sie auf „Weiter“.

6 Wählen Sie im Bildschirm „Einbetten“ die Einstellung „Eingebettetes Video“ als Symboltyp aus.

7 Klicken Sie im Videoimportbildschirm auf „Weiter“, um den Abschlussbildschirm aufzurufen und klicken Sie auf „Fertig stellen“, um das Video zu importieren.

   Im Bedienfeld „Bibliothek“ und auf der Bühne erscheint ein neues Videosymbol mit einer Verknüpfung zu der FLV-Datei.

8 Wählen Sie das Video-Objekt auf der Bühne aus, und geben Sie im Eigenschafteninspektor einen Instanznamen für das Video-Objekt ein.

   Anhand des Instanznamens lässt sich das Video-Objekt mit den Eigenschaften der Video-Klasse in ActionScript steuern.

Externe FLV-Dateien in einer Anwendung wiedergeben

Es ist keine externe FLV-Datei in eine SWF-Datei eingebettet. Externe FLV-Dateien befinden sich auf der Speicherkarte eines Geräts, auf einem Remote-Webserver oder auf einem Adobe Flash Media Server. Mit ActionScript können Sie externe FLV-Dateien in Flash Lite 3.x zur Laufzeit wiedergeben und steuern.

Die Wiedergabe externer FLV-Dateien bietet etliche Vorteile gegenüber der Einbettung von Video in Flash-Dokumente. Da externe FLV-Dateien weniger Speicher beanspruchen als eingebettete Videodateien, können Sie lange Videoclips verwenden, ohne dadurch die Wiedergabe zu verlangsamen. Externe FLV-Dateien können andere Bildraten aufweisen als das Flash-Dokument, in dem sie wiedergegeben werden. Sie können beispielsweise die Bildrate des Flash-Dokuments auf 30 Bilder pro Sekunde (BpS) und die des Videos auf 21 BpS einstellen. Dieses Steuerelement gewährleistet eine ruckfreie Videowiedergabe. Außerdem erlaubt das Steuerelement die Wiedergabe der FLV-Dateien in unterschiedlichen Bildraten, ohne dass der vorhandene Rich Media-Inhalt verändert werden muss.

Externe FLV-Dateien werden in Video-Objekten angezeigt. Video-Objekte können ebenso wie MovieClip-Objekte manipuliert werden. Video kann gedreht und skaliert werden. Der Alphatransparenzwert kann festgelegt werden. Das Video kann mit anderem Rich Media-Inhalt zusammengesetzt und vermischt werden. Mit der Sound-Klasse können Sie den Videosound steuern.

1 Wählen Sie aus der Bibliothek in Flash „Neues Video“, um ein Videosymbol zu erstellen. Vergewissern Sie sich, dass als Videotyp „Video (durch ActionScript gesteuert)“ gewählt wurde, und klicken Sie auf OK.

2 Ziehen Sie das Videosymbol auf die Bühne und geben Sie ihm einen Instanznamen. In diesem Beispiel wird der Instanzname **vo** verwendet.

3 Wählen Sie in der Zeitleiste Bild 1 und öffnen Sie das Bedienfeld „Aktionen“.

4 Geben Sie den folgenden Code ein :

```
// Create a NetConnection object
var nc:NetConnection = new NetConnection();
// Declare a NetStream object
var ns:NetStream;
/* When a change in connection status occurs,
the onStatus handler is called and passed an
infomation object with information about the change.
Use the information object to run code when
connection events occur.
*/
nc.onStatus = function(info) {
    // Output all connection status information
    trace(info.code);
    // Run the following code if the connection is successful.
    if(info.code == "NetConnection.Connect.Success"){
        // Create a NetStream object and pass it the NetConnection object
        ns = new NetStream(nc);
        // Attach the NetStream object to the video object on Stage.
        vo.attachVideo(ns);
        // Play the file "sample.flv". Do not include ".flv" when
        // playing the file on Flash Media Server.
        ns.play("sample");
    }
}
/* Call NetConnection.connect() to create a connection.
To connect to a local file or to a file on a web server, pass null.
To connect to Flash Media Server, pass the address to
an application on the server.
*/
nc.connect("rtmp://fmsexamples.adobe.com/vod");
```

Hinweis: *Um FLV-Dateien von einem Webserver laden zu können, müssen die Dateierweiterung und der MIME-Typ auf dem Webserver registriert werden. Der MIME-Typ für FLV-Dateien ist video/x-flv. Unter Umständen ist der MIME-Typ bereits beim Webserver registriert. Weitere Informationen finden Sie in der Dokumentation zum Webserver.*

FLV-Dateien mit ActionScript steuern

Verwenden Sie ActionScript zum Herstellen der Verbindung zu externen FLV-Dateien und zum Anzeigen und Steuern dieser Dateien. Mit der NetConnection-Klasse können Sie die Verbindung herstellen. Mit der Video-Klasse können Sie das Video anzeigen. Mit der NetStream-Klasse können Sie die Wiedergabe steuern.

Rufen Sie zur Steuerung der FLV-Dateien die Methoden der NetStream-Klasse auf, z. B. `NetStream.pause()`. Rufen Sie zur Steuerung des Gerätevideos die Methoden der Video-Klasse auf. Wenn Sie beispielsweise das Gerätevideo anhalten möchten, rufen Sie die Methode `Video.pause()` auf.

Eine vollständige Beschreibung dieser Klassen zusammen mit Codebeispielen für deren Verwendung finden Sie im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch.* Flash Lite 3.x unterstützt außerdem alle im *ActionScript 2.0-Referenzhandbuch* beschriebenen Methoden, Eigenschaften und Ereignisse, mit Ausnahme der Eigenschaft `MovieClipLoader.checkPolicyFile`.

Mit Gerätevideos arbeiten

Informationen zu Gerätevideos

Ab der Version 2.0 kann Flash Lite *Gerätevideo* wiedergeben, also alle Videoformate und Videokodierungen, die vom Zielgerät unterstützt werden. Die Geräte unterstützen unterschiedliche Video-Codecs und Videoformate. Zu den gebräuchlichen Gerätevideoformaten zählen unter anderem 3GP, 3G2 (oder 3GPP2) und MPEG-4. Flash Lite kann alle Videoformate abspielen, die das Zielgerät unterstützt.

Die Gerätevideos werden dabei von Flash Lite weder dekodiert noch direkt wiedergegeben. Stattdessen überlässt Flash Lite die Dekodierung und die Wiedergabe von Gerätevideo dem jeweiligen Gerät. Daher gelten für Gerätevideo folgende Beschränkungen:

- Gerätevideo kann nicht gedreht oder geneigt werden. Einige Geräte unterstützen das Skalieren.
- Gerätevideo kann nicht mit der Zeitleiste synchronisiert werden.
- Gerätevideo kann nicht mit anderen Medien zusammengesetzt oder gemischt werden. Das Gerät gibt Video über allen anderen Rich Media-Inhalten wieder.
- Die Lautstärke eines Videoclips kann nicht gesteuert werden.

Um Gerätevideo bereitzustellen, sollte es in einer Anwendung gebündelt oder von einer externen Datei auf dem Gerät oder von einem Netzwerkspeicherort aus geladen werden.

Siehe auch

„Unterstützte Gerätevideoformate ermitteln“ auf Seite 83

„Gerätevideo in eine Anwendung importieren“ auf Seite 84

„Externes Gerätevideo in einer Anwendung wiedergeben“ auf Seite 91

Unterstützte Gerätevideoformate ermitteln

Verschiedene Geräte unterstützen unterschiedliche Videokodierungen und Videoformate. Führen Sie einen der folgenden Schritte durch, um zu ermitteln, welche Formate ein Gerät unterstützt:

- Wählen Sie in Adobe Device Central die Registerkarte „Geräteprofile“. Weitere Informationen finden Sie in der Hilfe zu Device Central.
- Lesen Sie die technischen Daten des Geräteherstellers.
- Verwenden Sie die ActionScript-Eigenschaft `System.capabilities.videoMIMETypes`.

Die Eigenschaft `System.capabilities.videoMIMETypes` enthält ein Array der MIME-Videotypen, die vom Gerät unterstützt werden. Jedes Element in dem Array hat das Format `video/`*`video-typ`*. Der folgende Code zeigt beispielsweise alle MIME-Videotypen im Ausgabefenster an, die von dem Gerät unterstützt werden:

```
var mimeTypes = System.capabilities.videoMIMETypes;
trace(mimeTypes.toString());
```

Hinweis: *Wählen Sie „Ansicht“ > „Flash-Ausgabe“ > „Anzeigen“, um das Ausgabefenster in Adobe Device Central anzuzeigen.*

Der folgende Code ermittelt zunächst, ob ein Gerät die Wiedergabe von 3GPP-Video unterstützt, bevor eine Videodatei dieses Typs wiedergegeben wird:

```
if (System.capabilities.videoMIMETypes["video/3gpp"]) {
    my_video.play("movie.3gp");
}
```

Gerätevideo in eine Anwendung importieren

Sie können Gerätevideo durch Importieren in die Bibliothek eines Flash-Dokuments bündeln. Bei der Veröffentlichung der Anwendung bettet das Authoring-Tool die FLV-Datei in die veröffentlichte SWF-Datei ein.

Verwenden Sie beim Importieren der Videodatei in ein Flash-Dokument den Videoimportassistenten oder das Bedienfeld „Bibliothek“. Mit beiden Verfahren wird der Dokumentbibliothek ein Videosymbol hinzugefügt.

Videoimportassistenten verwenden

1 Wählen Sie in Flash „Datei“ > „Importieren“ > „Video importieren“.

2 Klicken Sie auf die Option zum Importieren einer Datei, die auf Ihrem Computer gespeichert ist. Klicken Sie dann auf „Durchsuchen“.

3 Wechseln Sie zu dem Ordner, der die gewünschte Gerätevideodatei enthält und wählen Sie sie aus.

   Falls die gewünschte Videodatei nicht in der Liste aufgeführt ist (oder nicht ausgewählt werden kann), wählen Sie „Alle Dateien“ (*.*) aus dem Popup-Menü mit den verfügbaren Dateitypen (Windows) oder wählen Sie „Alle Dateien“ aus dem Popup-Menü „Aktivieren“ (Macintosh).

4 Klicken Sie auf „Öffnen“.

5 Klicken Sie im Videoimportassistenten auf „Weiter“.

6 In Flash Lite 2.0-Anwendungen ist die einzige verfügbare Bereitstellungsoption „Als Mobilgerätvideo in SWF gebündelt“. Klicken Sie auf „Beenden“.

   Es wird ein Videosymbol im Bedienfeld „Bibliothek“ angezeigt. Dieses Symbol ist mit der Gerätevideodatei verknüpft.

7 Ziehen Sie das Video-Objekt aus der Bibliothek auf die Bühne, und geben Sie ihm im Eigenschafteninspektor einen Instanznamen.

Das Bedienfeld „Bibliothek“ zum Importieren von Gerätevideos verwenden

1 Öffnen Sie in Flash das Bedienfeld „Bibliothek“ („Fenster“ > „Bibliothek“).

2 Öffnen Sie das Popupmenü der Bibliothek und wählen Sie „Neues Video“ aus.

   Das Dialogfeld „Videoeigenschaften“ wird angezeigt.

3 Wählen Sie im Dialogfeld „Videoeigenschaften“ die Option aus, mit der das Quellvideo als Bundle in die SWF-Datei eingeschlossen wird. Klicken Sie dann auf „Importieren“.

4 Wechseln Sie zu dem Ordner, der die gewünschte Gerätevideodatei enthält und wählen Sie sie aus.

   Falls die gewünschte Videodatei nicht in der Liste aufgeführt ist (oder nicht ausgewählt werden kann), wählen Sie „Alle Dateien“ (*.*) aus dem Popup-Menü mit den verfügbaren Dateitypen (Windows) oder wählen Sie „Alle Dateien“ aus dem Popup-Menü „Aktivieren“ (Macintosh).

5 Klicken Sie auf „Öffnen“.

6 Wählen Sie im Dialogfeld „Videoeigenschaften“ die Option „Export für ActionScript“ und geben Sie eine Zeichenfolge in das Feld „Bezeichner“ ein.

7 Klicken Sie auf „OK“, um das Dialogfeld „Videoeigenschaften“ zu schließen.

Es wird ein Videosymbol im Bedienfeld „Bibliothek“ angezeigt. Dieses Symbol ist mit der Gerätevideodatei verknüpft.

Gebündelte Gerätevideos in einer Anwendung wiedergeben

Sie können diese Anwendung erstellen, indem Sie ein Video auf die Bühne ziehen und den Schaltflächen ActionScript hinzufügen, um das Video wiederzugeben oder zu stoppen.

1 Wählen Sie in Flash „Datei“ > „Neu“ > „Flash-Datei (Mobil)“ und klicken Sie auf OK.

2 Klicken Sie in Adobe Device Central auf „Erstellen“.

3 Wählen Sie in Flash „Datei“ > „Speichern“ und speichern Sie die Datei als bundled_video.fla ab.

4 Importieren Sie die Gerätevideodatei ocean.3gp aus dem Ordner Flash_Lite_2x\Samples\Video\. Weitere Informationen zum Importieren von Gerätevideos finden Sie unter „Gerätevideo in eine Anwendung importieren“ auf Seite 84.

   Hinweis: *Laden Sie von der Seite mit Beispielen und Tutorials unter (www.adobe.com/go/learn_flt_samples_and_tutorials_de) die komprimierte Datei für Ihre Flash Lite-Version herunter und dekomprimieren Sie die Datei.*



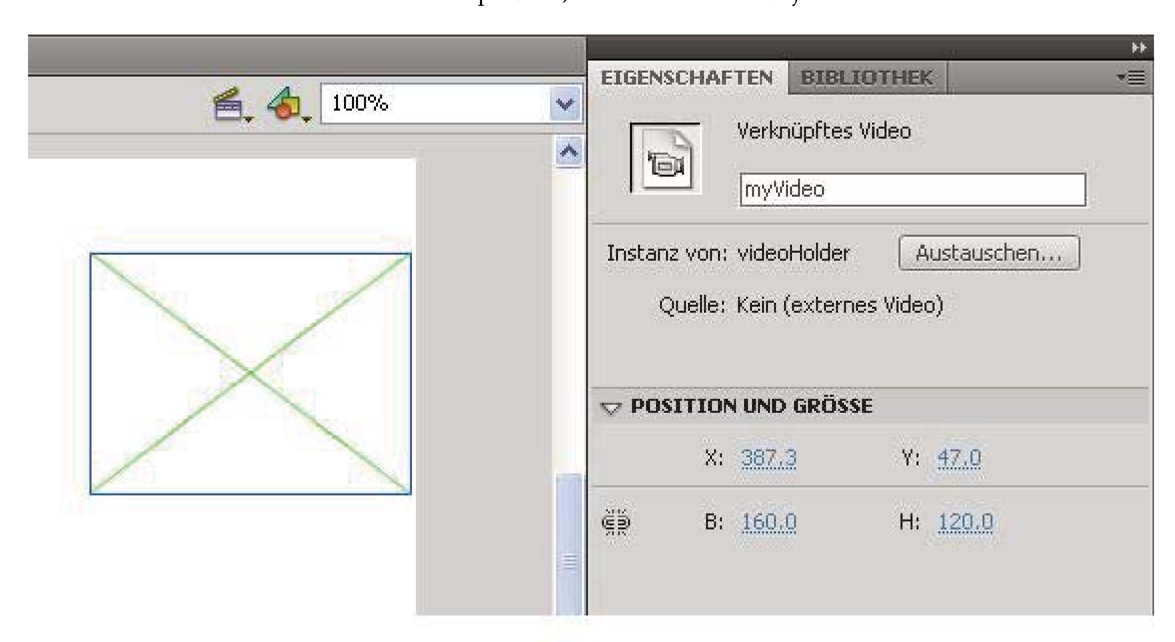
   Ein neues Videosymbol wird im Bedienfeld „Bibliothek“ angezeigt.



5 Ziehen Sie eine Instanz des Videosymbols auf die Bühne.

6 Wählen Sie das Video-Objekt auf der Bühne aus und geben Sie im Eigenschafteninspektor als Instanznamen **myVideo** ein.

Verwenden Sie diesen Namen im ActionScript-Code, um auf das Video-Objekt zu verweisen.



7 Stellen Sie im Eigenschafteninspektor die Breite auf 176 und die Höhe auf 144 ein.

   Diese Abmessungen entsprechen dem Quellvideo. Je nach Gerät werden Gerätevideos in Flash Lite nicht immer an die Größe des Begrenzungsrahmens angepasst.

8 Um Schaltflächen zur Videosteuerung hinzuzufügen, öffnen Sie die Bibliothek der vordefinierten Schaltflächen („Fenster" > „Allgemeine Bibliotheken" > „Schaltflächen").

9 Doppelklicken Sie im Bedienfeld „Bibliothek - Schaltflächen" auf den Ordner „Circle Buttons", um ihn zu öffnen.

10 Ziehen Sie eine Instanz des Schaltflächensymbols „Wiedergeben" auf die Bühne.

11 Ziehen Sie eine Instanz des Schaltflächensymbols „Anhalten" auf die Bühne.

12 Wählen Sie die Schaltfläche „Wiedergeben" auf der Bühne aus und fügen Sie durch Eingabe (oder Kopieren und Einfügen) folgenden Code in das Bedienfeld „Aktionen" ein:

```
on(press) {
    myVideo.play();
}
```

13 Wählen Sie die Schaltfläche „Anhalten" auf der Bühne aus und geben Sie folgenden Code in das Bedienfeld „Aktionen" ein:

```
on(press) {
    myVideo.stop();
}
```

14 Wählen Sie „Steuerung" > „Film testen", um den Film in Device Central zu testen.

Gebündeltes Gerätevideo zur Laufzeit in einer Anwendung wiedergeben

Ziehen den Platzhalter für das Video auf die Bühne, um die Anwendung zu erstellen. Fügen Sie den Schaltflächen ActionScript hinzu, um das Video wiederzugeben oder zu stoppen. Über die Schaltfläche „Wiedergeben“ wird das Video dynamisch aus dem Bedienfeld „Bibliothek“ abgerufen. Diese Anwendung verwendet nur ein Video. Eine Anwendung kann über eine beliebige Anzahl von Videodateien im Bedienfeld „Bibliothek“ verfügen, die zur Laufzeit in die Videoplatzhalter gezogen werden können.

1 Wählen Sie in Flash „Datei“ > „Neu“ > „Flash-Datei (Mobil)“ und klicken Sie auf OK.

2 Klicken Sie in Adobe Device Central auf „Erstellen“.

3 Wählen Sie in Flash „Datei“ > „Speichern“ und speichern Sie die Datei als bundled_video2.fla ab.

4 Importieren Sie die Gerätevideodatei ocean.3gp aus dem Ordner Flash_Lite_2x\Samples\Video\. Weitere Informationen zum Importieren von Gerätevideos finden Sie unter „Gerätevideo in eine Anwendung importieren“ auf Seite 84.

   Hinweis: *Laden Sie von der Seite mit Beispielen und Tutorials unter (www.adobe.com/go/learn_flt_samples_and_tutorials_de) die komprimierte Datei für Ihre Flash Lite-Version herunter und dekomprimieren Sie die Datei.*

   Ein neues Videosymbol wird im Bedienfeld „Bibliothek“ angezeigt.



5 Klicken Sie in der Bibliothek mit der rechten Maustaste (Windows) bzw. bei gedrückter Taste <Ctrl > (Macintosh) auf das Videosymbol ocean.3gp und wählen Sie im Kontextmenü die Option Eigenschaften aus. Das Dialogfeld „Videoeigenschaften“ wird angezeigt.

6 Aktivieren Sie im Dialogfeld „Videoeigenschaften“ die Option „Export für ActionScript“ und geben Sie **ocean_video** in das Feld „Bezeichner“ ein, wie in der folgenden Abbildung gezeigt:



7 Klicken Sie auf „OK“, um das Dialogfeld „Videoeigenschaften“ zu schließen.

8 So erstellen Sie den Platzhalter-Videoclip:

a Klicken Sie in der Titelleiste des Bedienfelds „Bibliothek“ auf die Schaltfläche für das Optionsmenü, und wählen Sie „Neues Video“ aus. Das Dialogfeld „Videoeigenschaften“ wird angezeigt.

b Geben Sie **videoHolder**in das Feld „Symbol“ ein.

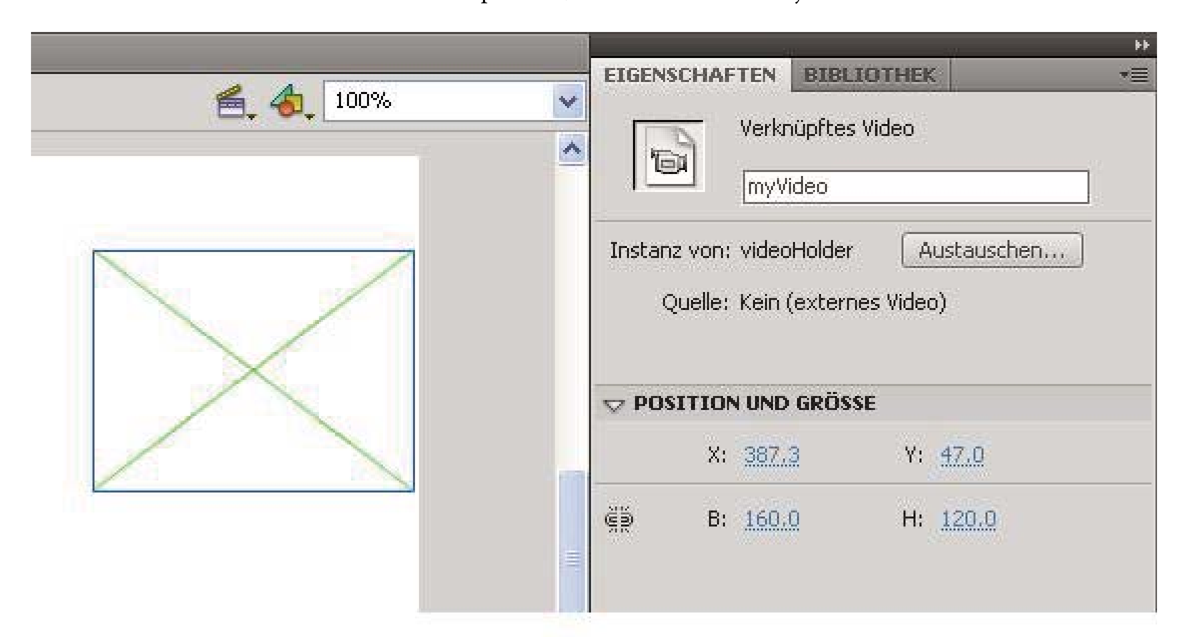


c Klicken Sie auf „OK“, um das Dialogfeld „Videoeigenschaften“ zu schließen.

9 Ziehen Sie eine Instanz des Videosymbols auf die Bühne.

10 Wählen Sie das Video-Objekt auf der Bühne aus und geben Sie im Eigenschafteninspektor als Instanznamen **myVideo** ein.

Verwenden Sie diesen Namen im ActionScript-Code, um auf das Video-Objekt zu verweisen.



11 Stellen Sie im Eigenschafteninspektor die Breite auf 176 und die Höhe auf 144 ein.

Diese Abmessungen entsprechen dem Quellvideo. Je nach Gerät werden Gerätevideos in Flash Lite nicht immer an die Größe des Begrenzungsrahmens angepasst.

12 Um Schaltflächen zur Videosteuerung hinzuzufügen, öffnen Sie die Bibliothek der vordefinierten Schaltflächen („Fenster“ > „Allgemeine Bibliotheken“ > „Schaltflächen“).

13 Doppelklicken Sie im Bedienfeld „Bibliothek - Schaltflächen“ auf den Ordner „Circle Buttons“, um ihn zu öffnen.

14 Ziehen Sie eine Instanz des Schaltflächensymbols „Wiedergeben“ auf die Bühne.

15 Ziehen Sie eine Instanz des Schaltflächensymbols „Anhalten“ auf die Bühne.

16 Wählen Sie die Schaltfläche „Wiedergeben“ auf der Bühne aus und fügen Sie durch Eingabe (oder Kopieren und Einfügen) folgenden Code in das Bedienfeld „Aktionen“ ein:

```
on(press) {
    myVideo.play("symbol://ocean_video");
}
```

17 Wählen Sie die Schaltfläche „Anhalten“ auf der Bühne aus und geben Sie folgenden Code in das Bedienfeld „Aktionen“ ein:

```
on(press) {
    myVideo.stop();
}
```

18 Wählen Sie „Steuerung“ > „Film testen“, um den Film in Device Central zu testen.

Externes Gerätevideo in einer Anwendung wiedergeben

Externe Gerätevideodateien befinden sich auf der Speicherkarte des Geräts oder auf einem Webserver. Wie für alle Gerätevideos gilt, dass das Gerät für die Dekodierung und das Anzeigen der Dateien verantwortlich ist.

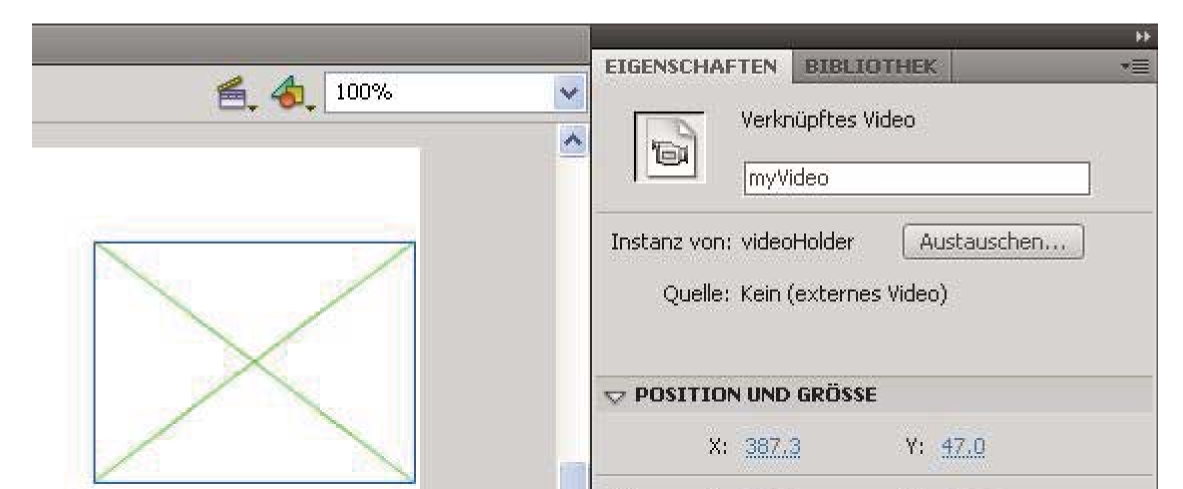
1. Wählen Sie in Flash „Datei“ > „Neu“ > „Flash-Datei (Mobil)“ und klicken Sie auf OK.
2. Klicken Sie in Adobe Device Central auf „Erstellen“.
3. Wählen Sie in Flash „Datei“ > „Speichern“ und speichern Sie die Datei als externe Datei namens external_device_video.fla ab.
4. Speichern Sie die Datei ocean.3gp in demselben Ordner ab wie die externe Datei external_device_video.fla.

   Diese Datei befindet sich im Ordner Flash_Lite_2x\Samples\Video\ im Paket mit Beispielen und Tutorials, das unterwww.adobe.com/go/learn_flt_samples_and_tutorials_de heruntergeladen werden kann.
5. Gehen Sie folgendermaßen vor, um das Videosymbol zum Anzeigen der externen Datei zu erstellen:
   a. Klicken Sie in der Titelleiste des Bedienfelds „Bibliothek“ auf die Schaltfläche für das Optionsmenü, und wählen Sie „Neues Video“ aus. Das Dialogfeld „Videoeigenschaften“ wird angezeigt.
   b. Geben Sie im Symbolfeld **Videoanzeige** ein.

      Dieser Name wird im Bedienfeld „Bibliothek“ verwendet.
   c. Wählen Sie „Video (durch ActionScript gesteuert)“ und klicken Sie auf OK.

      Achten Sie darauf, dass die Option „Quelle in SWF für mobil und Gerät bündeln“ nicht markiert ist.
6. Ziehen Sie eine Instanz des Videosymbols auf die Bühne.
7. Wählen Sie das Video-Objekt auf der Bühne aus und geben Sie im Eigenschafteninspektor als Instanznamen **myVideo** ein.

Verwenden Sie diesen Namen im ActionScript-Code, um auf das Video-Objekt zu verweisen.



8 Stellen Sie im Eigenschafteninspektor die Breite des Video-Objekts auf 176 und die Höhe auf 144 ein.

   Diese Abmessungen entsprechen dem Quellvideo. Je nach Gerät werden Gerätevideos in Flash Lite nicht immer an die Größe des Begrenzungsrahmens angepasst.

9 Um Schaltflächen zur Videosteuerung hinzuzufügen, öffnen Sie die Bibliothek der vordefinierten Schaltflächen („Fenster“ > „Allgemeine Bibliotheken“ > „Schaltflächen“).

10 Doppelklicken Sie im Bedienfeld „Bibliothek - Schaltflächen“ auf den Ordner „Circle Buttons“, um ihn zu öffnen.

11 Ziehen Sie eine Instanz des Schaltflächensymbols „Wiedergeben“ auf die Bühne.

12 Ziehen Sie eine Instanz des Schaltflächensymbols „Anhalten“ auf die Bühne.

13 Wählen Sie die Schaltfläche „Wiedergeben“ auf der Bühne aus und fügen Sie durch Eingabe (oder Kopieren und Einfügen) folgenden Code in das Bedienfeld „Aktionen“ ein:

```
on(press) {
    myVideo.play("ocean.3gp");
}
```

   Zum Wiedergeben einer externen Videodatei übergeben Sie den relativen oder absoluten Dateipfad oder die URL der Videodatei an die Methode `Video.play()`. Im folgenden Beispiel befindet sich die SWF-Datei im selben Ordner des Geräts wie die 3GP-Datei. Sie können auch wie im folgenden Beispiel gezeigt einen relativen Pfad für die SWF-Datei angeben:

```
myVideo.play("folder1/folder2/ocean.3gp");
```

   Je nach Gerät kann auch das Protokoll `file://` verwendet werden, um eine Videodatei an einem bestimmten Speicherort wiederzugeben:

```
myVideo.play("file://c:/folder1/folder2/ocean.3gp");
```

Hinweis: *Nicht alle Geräte unterstützen das Protokoll* `file://`*. Wenn Sie sich für die Verwendung dieses Protokolls entscheiden, müssen Sie die Anwendung auf allen Zielgeräten testen.*

Einige Geräte unterstützen das Laden von Videodateien von einer Netzwerkadresse mit RTSP (Real Time Streaming Protocol). Das Streaming auf HTTP-Basis wird nicht unterstützt. Im folgenden Beispiel wird eine 3GPP-Datei geladen und wiedergegeben:

```
myVideo.play("rtsp://www.example.com/video/ocean.3gp");
```

14 Wählen Sie die Schaltfläche „Anhalten“ auf der Bühne aus und geben Sie folgenden Code in das Bedienfeld „Aktionen“ ein:

```
on(press) {
    myVideo.stop();
}
```

15 Wählen Sie „Steuerung“ > „Film testen“, um den Film in Device Central zu testen.

Eigenschaften von Gerätevideosymbolen anzeigen und bearbeiten

Im Dialogfeld Videoeigenschaften können Sie die Videosymbole, die sich im Bedienfeld Bibliothek befinden, anzeigen und bearbeiten.

Öffnen Sie das Dialogfeld „Videoeigenschaften“.

Führen Sie eine der folgenden Aktionen aus:

- Klicken Sie in der Bibliothek mit der rechten Maustaste (Windows) bzw. bei gedrückter Taste „Ctrl“ (Macintosh) auf ein Videosymbol und wählen Sie im Kontextmenü die Option „Eigenschaften“ aus.
- Wählen Sie im Bedienfeld „Bibliothek“ ein Videosymbol aus und wählen Sie dann im Optionsmenü in der Titelleiste die Option „Eigenschaften“ aus.

Gerätevideo in ein Videosymbol importieren

1 Wählen Sie im Bedienfeld „Bibliothek“ ein Videosymbol aus und öffnen Sie das Dialogfeld „Videoeigenschaften“.

2 Wählen Sie im Dialogfeld „Videoeigenschaften“ die Option aus, mit der das Quellvideo als Bundle in die SWF-Datei eingeschlossen wird.

3 Klicken Sie auf „Importieren“. Suchen Sie dann im Dateibrowser die zu importierende Gerätevideodatei und wählen Sie sie aus.

   Falls die gewünschte Videodatei nicht in der Liste aufgeführt ist (oder nicht ausgewählt werden kann), wählen Sie „Alle Dateien“ (*.*) aus dem Popup-Menü mit den verfügbaren Dateitypen (Windows) oder wählen Sie „Alle Dateien“ aus dem Popup-Menü „Aktivieren“ (Macintosh).

4 Klicken Sie auf „Öffnen“, um den Dateibrowser zu schließen.

5 Klicken Sie auf „OK“, um das Dialogfeld „Videoeigenschaften“ zu schließen.

Hinweis: *Die Schaltfläche „OK“ ist erst verfügbar, wenn Sie ein Gerätevideo importieren oder die Option deaktivieren, mit der das Quellvideo als Bundle in die SWF-Datei eingeschlossen wird.*

Einem Videosymbol einen Bezeichner zuweisen

1 Wählen Sie das Videosymbol aus und öffnen Sie das Dialogfeld „Videoeigenschaften“.

2 Importieren Sie ein Gerätevideo in das Symbol, falls noch nicht geschehen.

3 Aktivieren Sie die Option „Export für ActionScript“.

4 Geben Sie in das Feld „Bezeichner“ einen Bezeichner für das Videosymbol ein.

Wie alle Bezeichner in ActionScript muss auch dieser Bezeichner mit einem Buchstaben, einem Unterstrich (_) oder einem Dollarzeichen ($) beginnen. Bei jedem weiteren Zeichen kann es sich um eine Ziffer, einen Buchstaben, einen Unterstrich oder ein Dollarzeichen handeln.

5 Klicken Sie auf „OK“, um das Dialogfeld „Videoeigenschaften“ zu schließen.

Gerätevideo mithilfe von ActionScript steuern

Zum Steuern der Videowiedergabe wird das Video-Objekt von ActionScript verwendet. Folgende Methoden des Video-Objekts stehen in Flash Lite 2.0 und höheren Versionen zur Verfügung:

- `Video.play()`
- `Video.stop()`
- `Video.pause()`
- `Video.resume()`
- `Video.close()`
- `Video.attachVideo()` (Flash Lite 3.0 und höher)

Hinweis: *Diese Methoden des Video-Objekts entsprechen ungefähr denselben Methoden, die im NetStream-Objekt der Desktopversion von Flash Player verfügbar sind. Diese Methoden des Video-Objekts stehen in der Desktop-Version von Flash Player nicht zur Verfügung.*

Die folgenden Methoden und Eigenschaften des Video-Objekts werden in Flash Lite 2.0 und höher nicht unterstützt:

- `Video.clear()`
- `Video.deblocking`
- `Video.height`
- `Video.smoothing`
- `Video.width`
- `Video._visible`

Weitere Informationen zur Verwendung des Video-Objekts in Flash Lite finden Sie im Eintrag zum Objekt „Video“ im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.

Kapitel 8: Flash Lite-Anwendungen für BREW entwickeln

Sie können Adobe Flash Lite 2.x-Anwendungen entwickeln, die auf Geräten ausgeführt werden können, die die Plattform „Binary Runtime Environment for Wireless“ (BREW) von QUALCOMM Incorporated verwenden. Dieses Kapitel richtet sich an die Entwickler, der zwar mit Flash Lite vertraut ist, aber noch nicht über genügend Kenntnisse der BREW-Plattform und deren Anforderungen verfügen, um voll funktionsfähige Anwendungen für BREW-Geräte zu entwickeln. Außerdem richtet es sich an Benutzer, die bereits Erfahrungen mit anderen Technologien zur Anwendungsentwicklung gesammelt haben und nun die Möglichkeiten von Flash Lite zur Entwicklung von BREW-Anwendungen erforschen möchten.

BREW-Grundlagen

BREW

BREW, die Plattform für Mobilfunkgeräte von QUALCOMM, wird weltweit von verschiedenen Herstellern auf einer Reihe von Geräten installiert und unterstützt. Adobe Flash CS4 Professional beinhaltet eine Sammlung von Tools und Ressourcen, mit denen Entwickler Flash Lite-basierte Anwendungen und Inhalte erstellen können, die von Benutzern auf BREW-konforme Geräte heruntergeladen und dort verwendet werden können.

Mit dem Flash-Authoring-Tool können Flash-Entwickler Flash Lite-Inhalte und -Anwendungen für BREW-konforme Geräte erstellen. Fertige Anwendungen legen die Entwickler dann den National Software Testing Laboratories (NSTL) zum True BREW Testing (TBT) vor. Nachdem die Anwendungen das TBT erfolgreich durchlaufen haben, werden sie an das BREW Delivery System (BDS) hochgeladen. Anwendungen, die ein Netzbetreiber im BDS zur Verteilung auswählt, werden auf dem Application Download Server (ADS) des Netzbetreibers zum Download angeboten. Benutzer BREW-konformer Geräte können diese Anwendungen dann herunterladen und kaufen. Sie profitieren dabei von den Funktionen der BREW-Plattform für die mobilfunkgestützte Verteilung und Rechnungsstellung über das BDS.

Entwickler, die mit BREW experimentieren möchten, dabei aber die Kosten scheuen, die für die Vollversion der BREW-Entwicklungsumgebung entstehen, können beispielsweise auf die Registrierung als authentifizierter BREW-Entwickler verzichten und nur die kostenlosen SDK (Software Development Kit)-Komponenten herunterladen. Schon diese Komponenten reichen aus, um eine Anwendung mit dem BREW-Simulator zu generieren und zu testen; das Hochladen und Testen einer Anwendung auf einem BREW-konformen Gerät oder das Simulieren bestimmter Geräteeigenschaften mit anpassbaren Gerätepacks ist jedoch nicht möglich.

BREW kann als ein „Ökosystem“ zur Unterstützung der Entwicklung und Bereitstellung von Inhalten für Mobilgeräte betrachtet werden. Das Ziel von BREW besteht darin, Entwicklern das Erstellen, Verteilen und Nutzen von Anwendungen für Mobilgeräte zu vereinfachen. Dieses Ökosystem setzt sich aus den folgenden Teilnehmergruppen zusammen:

Entwickler verwenden die Tools des BREW SDK, um Inhalte mit C++ zusammen mit APIs oder mit einer BREW-konformen Erweiterung zu entwickeln. Das BREW SDK kann kostenlos heruntergeladen werden; Sie müssen aber sich bei QUALCOMM registrieren, um Zugriff auf die Tools Suite und die Tools im SDK zu erhalten.

National Software Testing Laboratories (NSTL) testet die von Entwicklern vorgelegten Anwendungen. Wenn diese Anwendungen die Anforderungen erfüllen, können sie im BDS bereitgestellt werden, andernfalls haben die Entwickler die Möglichkeit, ihre Anwendungen zu überarbeiten und dann erneut vorzulegen.

QUALCOMM verwaltet das BREW Delivery System (BDS), ein webbasiertes System, das die im QUALCOMM-Kataloge aufgeführten Inhalte zusammen mit Informationen zu Preisen und Rechnungsstellung bereitstellt. Die Preisgestaltung kann verschiedene Modelle enthalten, so auch eine Betreiberauktion.

Betreiber (Netzbetreiber) verwenden den Application Download Server (ADS), um ihren Kunden ausgewählte Inhalte über eine mobilfunkgestützte Bereitstellung (Over the Air / OTA) zu Wiederverkaufspreisen anzubieten.

Gerätehersteller verwenden den BREW-Client, der allgemeine Anwendungsprogrammierschnittstellen (APIs) zur standardisierten Entwicklung von mobilfunkgestützten Anwendungen offenlegt. Darüber hinaus umfasst der Client einen Anwendungsmanager, den Benutzer erwerben und zur Verwaltung von BREW-Anwendungen einsetzen können.

Flash Lite und BREW-Entwicklungstools

Zum Entwickeln und Testen von Flash Lite-Anwendungen für die BREW-Umgebung benötigen Sie die folgende Hardware und Software:

Hardware

Windows-Computer Informationen zu den Systemanforderungen für BREW SDK und BREW Tools Suite finden Sie im Abschnitt „System Requirements" im Handbuch *Starting with BREW* (verfügbar auf der BREW-Website). Informationen zu den Systemanforderungen für Adobe-Software finden Sie unter www.adobe.com/go/sysreqs_de.

BREW-konforme Geräte mit gesetztem Testbit Informationen zum Erwerben, Aktivieren und Freigeben von BREW-Geräten für Tests finden Sie im Abschnitt „Handset Acquisition and Readiness" im Handbuch *Getting Started with BREW* (verfügbar auf der BREW-Website). Geräte, die BREW 2.x unterstützen, müssen zur Testfreigabe an QUALCOMM gesendet werden, BREW 3.x-Geräte sind vorab freigegeben.

Datenkabel Datenkabel sind erforderlich, um Flash-Anwendungen an die BREW-Geräte hochzuladen, für die Sie Anwendungen entwickeln. Bei den meisten Geräten sind die Datenkabel bereits im Lieferumfang enthalten, andernfalls lesen Sie den Abschnitt „Acquire BREW Handsets and Data Cables" im Handbuch *Getting Started with BREW* (verfügbar auf der BREW-Website).

Software

BREW SDK und BREW Tools Suite Informationen zur Installation des BREW SDK und der BREW Tools Suite finden Sie in Kapitel 2 „Installing the BREW SDK" im Handbuch *Getting Started with BREW* (verfügbar auf der BREW-Website). Sie müssen sich als BREW-Entwickler authentifizieren lassen, um die BREW Tools Suite herunterladen und installieren zu können. Für das BREW SDK ist dies nicht erforderlich. Informationen zur Registrierung finden Sie auf der BREW-Website.

USB-Treiber für Zielgeräte Installieren Sie die USB-Treiber für die beabsichtigten Zielgeräte. Wie Sie die erforderlichen Treiber erhalten können, erfahren Sie vom Hersteller des jeweiligen Geräts.

Flash Lite-Erweiterung für BREW-konforme Geräte. Diese Geräte sind auf der BREW-Website nur zu Testzwecken verfügbar. Die Erweiterung wird automatisch auf die Geräte der Benutzer heruntergeladen, wenn sie eine Flash Lite für BREW-Anwendung auswählen und herunterladen.)

Diese Kompontenten werden unter „System für BREW einrichten" auf Seite 99 genau beschrieben.

In BREW unterstützte Flash Lite-Funktionen

Flash Lite für BREW unterstützt einen Teil der Funktionen, die sowohl in Flash Lite 2.0 als auch in Flash Lite 2.1 zur Verfügung stehen. Die von Flash Lite für BREW unterstützten Funktionen sind im Folgenden aufgeführt. Weitere Informationen zu diesen Funktionen finden Sie unter „Flash Lite 3.0-Funktionen“ auf Seite 7.

- Basiert auf Adobe Flash Player 7 von Adobe, unterstützt Adobe ActionScript 2.0
- Verarbeitung von XML-Daten
- Verwaltung von permanenten Daten
- Gerät, Vektorschriften
- Gerätesound für Ereignisse
- Inline-Video (beschränkt auf die vom Gerät unterstützten Videoformate)
- Streaming-Video
- XML-Sockets
- Netzwerkzugang und HTTP-Streaming
- Unterstützung von Inlinetext (beschränkt auf 64.000 Zeichen mit weiteren Einschränkungen durch individuelle Geräteimplementationen. Einige Geräte unterstützen nur 1000 Zeichen.)
- Dynamisches Multimedia
- Unterstützung von Tastenfeld und Tastatur

In BREW nicht unterstützte Flash Lite-Funktionen

Die Hersteller von BREW-Geräten können die Verwendung bestimmter Flash Lite-Funktionen einschränken. Dies bedeutet, dass diese Funktionen nicht mit dem Gerät verwendet werden können. Informationen zu bestimmten Geräten finden Sie im Developer Resources-Bereich auf der BREW-Website.

Neben den Funktionen, die auf einem bestimmten Gerät nicht unterstützt werden, enthält die folgende Liste die Funktionen, die in der aktuellen Flash Lite BREW-Implementierung nicht unterstützt werden (und somit auf keinem Gerät zur Verfügung stehen):

Flash Lite-Sounddekodierung Alle Soundformate werden unverändert zur Wiedergabe an das Gerät übergeben. Wenn der Sound beispielsweise im MP3-Format vorliegt und das Gerät MP3 unterstützt, wird der Sound wiedergegeben. Flash Lite dekodiert keine Sounds intern in BREW, wie dies auf anderen Plattformen erfolgt.

Streaming-Sound Alle Sounds müssen vollständig geladen sein, bevor sie wiedergegeben werden können. Die aktuelle BREW-Implementierung unterstützt keine progressive Wiedergabe von Sounds während des Ladens über ein Netzwerk.

ActionScript-Befehle Die folgenden ActionScript-`fscommand2()`-Funktionen stehen auf jedem BREW-Gerät zur Verfügung:

`ExtendBacklightDuration()`	`GetNetworkConnectStatus()`
`GetBatteryLevel()`	`GetNetworkName()`
`GetMaxBatteryLevel()`	`GetNetworkRequestStatus`
`GetMaxSignalLevel()`	`GetPowerSource()`
`GetNetworkConnectionName()`	`GetSignalLevel()`

Die folgende ActionScript-`fscommand()`-Funktion steht auf keinem BREW-Gerät zur Verfügung:

fscommand(Launch)	

Unterstützung von getURL() Einige BREW-Handsets unterstützen je nach Implementierung bestimmte `getURL()`-Funktionen. MMS (Multimedia Messaging Service) und HTTP werden grundsätzlich nicht unterstützt (es ist keine API vorhanden, die es Flash Lite ermöglicht, einen Browser zu starten).

Hinweis: *`mailto`-Anforderungen werden nicht direkt unterstützt. Alternativ können Sie das Protokoll Short Message Service (SMS) verwenden, bei dem die maximale Nachrichtenlänge jedoch auf 160 Zeichen beschränkt ist.*

Wallpaper-Inhalt Flash Lite-Wallpaper werden von der Flash Lite für BREW-Implementierung nicht unterstützt.

Animierte Klingeltöne Animierte Klingeltöne werden von der Flash Lite für BREW-Implementierung nicht unterstützt.

Unterstützte BREW SDK-Versionen

Flash Lite 2.1 für BREW-Geräte unterstützt derzeit BREW-Versionen ab 2.1.3. Die auf einem bestimmten Gerät ausgeführte BREW-Version finden Sie in den technischen Daten zum Gerät auf der BREW-Website. Die zwei Geräte, die Flash Lite für BREW derzeit unterstützen (das Samsung SCH-A950 und das LG VX9800), arbeiten mit unterschiedlichen Versionen von BREW (2.x bzw. 3.x). Da die BREW SDKs abwärtskompatibel sind, können Sie das SDK und die Tools für die aktuellste Version herunterladen (zum Zeitpunkt der Drucklegung 3.x). Sie müssen nicht die Version verwenden, die von dem Gerät unterstützt wird, für das Sie Anwendungen schreiben. Wenn Sie also Anwendungen für das Samsung SCH-A950 entwickeln (ein 2.x-Gerät), können Sie trotzdem die 3.x-Version des BREW SDK und der Tools Suite herunterladen und verwenden.

Die Unterschiede zwischen BREW 2.x und 3.x hinsichtlich der Benutzerschnittstelle sind nur minimal. Flash Lite verhält sich auf beiden Plattformen gleich. Die wichtigsten Unterschiede liegen darin, wie Dateien auf dem Gerät gespeichert und Anwendungen in der BREW Tools Suite geladen werden. Weitere Informationen zur Dateisystemstruktur finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.

Unterstützte Geräte

Die BREW-Plattform wird derzeit von den folgenden Geräten unterstützt:

Gerät	Unterstützte Version
Samsung SCH-A950	BREW-Version 2.1.3
LG VX9800	BREW-Version 3.1.2

Das Adobe-Kundendienstteam wird weitere Geräte zertifizieren, sobald diese verfügbar werden. Eine aktuelle Liste der unterstützten Geräte finden Sie unter www.adobe.com/go/mobile_supported_devices/.

Weitere Informationsquellen

Informationen zu BREW finden Sie auf der BREW-Website. Der Developer Home-Bereich der Website enthält Informationen zum BREW SDK, den BREW-Tools und –Utilities sowie zu den Eigenschaften bestimmter Geräte, die BREW unterstützen.

Hinweis: *Sie müssen sich als BREW-Entwickler registrieren lassen, um die BREW Tools Suite herunterladen und installieren zu können oder um Ihre Geräte zum Testen der BREW-Konformität freizugeben. Informationen zur Registrierung finden Sie auf der Website.*

System für BREW einrichten

Arbeitsablauf beim Einrichten Ihres Systems für BREW

Zum Erzeugen von Flash Lite-Dateien für BREW müssen Sie entsprechende Software von verschiedenen Quellen auf Ihrem Computer installieren. Außerdem muss die erforderliche Hardware vorhanden sein (BREW-konforme Geräte, Datenkabel usw.). Viele spezifische Informationen zu Installation und Konfiguration dieser Komponenten finden Sie in anderen Flash Lite- oder BREW-Dokumenten.

Sie müssen die folgenden Schritte ausführen, um Ihr System so einzurichten, dass es das Flash Lite-Authoring für BREW unterstützt:

1. Als BREW-Entwickler registrieren
2. BREW SDK und BREW Tools Suite installieren
3. Adobe Flash Lite 2.1 für BREW installieren
4. USB-Treiber für die Zielgeräte installieren

Als BREW-Entwickler registrieren

Die Registrierung als BREW-Entwickler erfolgt auf QUALCOMMs BREW-Hauptseite. Folgen Sie den Links auf der Developer-Homepage, um sich als authentifizierter BREW-Entwickler registrieren zu lassen.

BREW SDK und BREW Tools Suite installieren

Im Folgenden wird die Installation des BREW SDK auf Ihrem Computer beschrieben. Dabei wird davon ausgegangen, dass Sie bereits als BREW-Entwickler registriert sind. Ausführliche Informationen zur Installation des BREW SDK oder der Tools Suite finden Sie unter „Installing the BREW SDK“ im Handbuch *Starting with BREW*. Rufen Sie die BREW-Website auf und wählen Sie dann Developer Home > BREW Documentation > Application Development Documentation.

1. Klicken Sie auf der BREW Developer-Homepage im Bereich „BREW Developer Home“ auf „Download BREW Tools“, um die Seite „BREW Development and Commercialization Tools“ aufzurufen.
2. Wechseln Sie zum Bereich „BREW Development „ und klicken Sie auf den Link zur aktuellsten BREW SDK-Version (zum Zeitpunkt der Drucklegung BREW SDK 3.1).

   Folgen Sie den Anweisungen auf dem Bildschirm, um das SDK herunterzuladen und zu installieren.
3. Installieren Sie zunächst das SDK, dann die Tools.

   Wechseln Sie in den Bereich „BREW Commercialization“ der Seite „BREW Development and Commercialization Tools“ und klicken Sie auf den Link zu den „BREW Testing and Commercialization Utilities“. Folgen Sie den Anweisungen auf dem Bildschirm, um die Tools Suite auf Ihrem Computer zu installieren.

   Ihr Startmenü sollte nun Einträge für BREW SDK und BREW Tools Suite enthalten.

Adobe Flash Lite 2.1 für BREW-Geräte installieren

Flash Lite 2.1 für BREW-Geräte umfasst drei Komponenten:

- Flash Lite Publisher für BREW (verfügbar unter www.adobe.com/go/support_flashlite_de)
- Flash Lite 2.1 für BREW (verfügbar unter www.adobe.com/go/support_flashlite_de)
- Flash Lite-Erweiterung für BREW-konforme Geräte (auf der BREW-Website, nur für Testzwecke. Die Erweiterung wird automatisch auf die Geräte der Benutzer heruntergeladen, wenn sie eine Flash Lite für BREW-Anwendung auswählen und herunterladen.)

USB-Treiber für Zielgeräte installieren

Jedes Gerät, für das Sie BREW-Anwendungen entwickeln möchten, verfügt über einen eigenen Satz von USB-Treibern, die auf Ihrem Computer installiert sein müssen. Neuere Geräte werden meist mit einer CD-ROM ausgeliefert, auf der diese Treiber enthalten sind. Wenn möglich, verwenden Sie diese Versionen. Zur Installation der USB-Treiber für ein Gerät befolgen Sie das Standardverfahren zur Installation von Gerätetreibern unter Windows.

Weitere Informationsquellen

Weitere Informationen zur Entwicklung und den Testverfahren von Flash Lite-Anwendungen finden Sie in den anderen Abschnitten dieses Dokuments. Auch die Onlinehilfe von Flash CS4 Professional enthält ausführliche Informationen zur Entwicklung von Anwendungen mit Flash Lite.

Weitere Informationen zu BREW finden Sie auf der BREW-Website von QUALCOMM.

Auch in den Hilfedateien des SDK finden Sie weiterführende Informationen zu den BREW-Entwicklungsprozessen und –Tools.

Zum Zeitpunkt der Drucklegung umfasst die BREW-Dokumentation zwei ähnlich benannte Dokumente, die beide zum Erlernen der Grundlagen von BREW geeignet sind. Diese Dokumente sind:

- *Getting Started with BREW* (eine zweiseitige Zusammenfassung der Hardware und Software, die Sie zur Entwicklung von BREW-Anwendungen benötigen).
- *Starting with BREW* (ein umfangreiches Dokument, in dem alle Informationen enthalten sind, die Sie zur Entwicklung von BREW-Anwendungen benötigen).

Flash Lite-Dateien für BREW erstellen

Arbeitsablauf beim Erstellen von Flash Lite-Dateien für BREW

Die Schritte beim Erstellen von Dateien für BREW sind denen beim Erstellen von Dateien für Flash Lite sehr ähnlich, außer dass Sie Ihre Anwendungen an die Eigenschaften des jeweiligen BREW-konformen Handsets anpassen müssen, für das Sie Anwendungen entwickeln. In diesem Abschnitt wird beschrieben, wie Sie mit Flash Inhalte für Flash Lite erstellen, die später mithilfe von Flash Lite Publisher für BREW (siehe Beschreibung in „Flash Lite-Dateien für BREW veröffentlichen“ auf Seite 103) für BREW veröffentlicht werden.

Die Entwicklung von Flash Lite-Inhalten für die BREW-Plattform geht immer nach dem gleichen Schema vonstatten und umfasst die folgenden Schritte:

Zielgeräte und Flash Lite-Inhaltstyp identifizieren

Zum Zeitpunkt der Drucklegung unterstützen die folgenden beiden Geräte die Plattform BREW: Samsung SCH-A950, das die BREW Version 2.1.3-Plattform unterstützt und LG VX9800, das die BREW Version 3.1.2-Plattform unterstützt. Diese Geräte weisen vollständig unterschiedliche Eigenschaften auf (beispielsweise besitzt das LG im Gegensatz zum Samsung eine QWERTY-Tastatur), daher ist es wichtig, dass Sie die Entwicklung Ihrer Anwendung auf die Fähigkeiten des jeweiligen Gerätes abstimmen.

Sie können zwei unterschiedliche Inhaltstypen erstellen (Anwendungen und Bildschirmschoner). Für jeden Typ gelten andere Anforderungen an das Anwendungsdesign. Informationen darüber, wie sich Bildschirmschoner von Anwendungen unterscheiden, finden Sie auf der BREW-Website. Rufen Sie den Bereich Developer FAQs > BREW Tools > MIF Settings > Screensaver auf.

Hinweis: *Derzeit unterstützt nur das Samsung SCH-A950 Inhalte des Typs „Bildschirmschoner“.*

Anwendung in Flash erstellen und testen

Der Lieferumfang von Adobe Flash CS4 Professional umfasst einen Adobe Device Central CS4-Emulator, mit dem Sie Ihre Anwendung testen können, ohne sie auf ein Gerät übertragen zu müssen. Mit diesem Device Central-Emulator können Sie das allgemeine Anwendungsdesign optimieren und sämtliche Probleme beheben, bevor Sie die Anwendung auf einem Mobilgerät testen.

Hinweis: *Das Testen in Flash mithilfe des Emulators unterscheidet sich vom Testen, das Sie mit dem Simulator aus dem BREW SDK durchführen. Die Abläufe beim Testen im Flash-Emulator werden in diesem Abschnitt, die Abläufe beim Testen im BREW-Simulator unter* *„Flash Lite-Dateien für BREW veröffentlichen“ auf Seite 103* *beschrieben. Da der Device Central-Emulator derzeit keine Gerätepacks für BREW-Geräte bietet, können mit dem Device Central-Emulator nur die allgemeinen Funktionen getestet werden. Mit dem BREW-Simulator testen Sie die Funktionen Ihrer Anwendung in der BREW-Umgebung.*

Zielgerät und Inhaltstyp identifizieren

Die BREW Developer-Homepage von QUALCOMM enthält weiterführende Informationen zu den besonderen Eigenschaften aller Geräte, die derzeit eine Version von BREW unterstützen. Einige Informationen beziehen sich auf Geräte in der Entwicklungsphase, die noch nicht auf dem Markt erhältlich sind. Anhand dieser Informationen können Sie ermitteln, welche Mobilteile als Zielgeräte für Ihre BREW-Anwendung geeignet sind. Das BREW SDK umfasst einige Beispielanwendungen mit verschiedenen Gerätepacks, so dass Sie mehr über die Arbeitsweise bestimmter Geräte erfahren können.

Anwendung in Flash erstellen

Der Prozess zum Erstellen von Flash Lite-Anwendungen für BREW ähnelt dem zum Erstellen einer allgemeinen Flash Lite-Anwendung. Mit Ausnahme der von BREW-Geräten nicht unterstützten Funktionen (siehe „In BREW nicht unterstützte Flash Lite-Funktionen“ auf Seite 97) und den BREW-spezifischen Authoring-Hinweisen (siehe unten) entwickeln Sie Anwendungen für BREW-Geräte, indem Sie die Entwicklungsschritte befolgen, die in der weiterführenden Flash-Dokumentation enthalten sind.

Die Informationen in den folgenden Themen gelten nur für das Flash Lite-Authoring für BREW.

Einen Browser starten

Die BREW-Handsets, für die Sie Anwendungen erstellen können, verwenden den Browser OpenWave, der keinen Hook zum Starten des Browsers für die BREW-Anwendung enthält. Wenn Sie also eine HTTP URL in den Funktionen `getURL()`, `loadVars(send)` oder `XML.send()` angeben, wird kein Browser aufgerufen. Stattdessen müssen Sie die Funktion `loadVars.sendAndLoad()` verwenden, um Informationen an einen Server zu senden und die Wiedergabe der SWF-Datei fortzusetzen, ohne ein neues Fenster zu öffnen oder den Inhalt eines vorhandenen Fensters oder Frames zu ersetzen.

Standardverhalten der Softtasten

Informationen und Beispiele, wie das Standardverhalten der Softtasten in Flash Lite programmiert wird, finden Sie unter „Softtasten verwenden“ auf Seite 51. In der folgenden Tabelle ist das Standardverhalten der Softtasten in Flash Lite 2.1 für BREW beschrieben.

Inhaltaktion	Bildschirmmodus	Aktion der linken Taste	Aktion der rechten Taste
Standard (Inhalt ruft `SetSoftKeys`-Befehl nicht auf)	Vollbildmodus	Zeigt Optionsmenü an	Beendet den Player
Nicht-Vollbildmodus (Inhalt ruft `SetSoftKeys` nicht auf)	Nicht-Vollbildmodus	Zeigt Optionsmenü an	Beendet den Player
Inhalt deaktiviert Softtasten	-	Keine	Keine
Inhalt setzt eine oder beide Softtasten außer Kraft (ruft `SetSoftKeys` auf)	-	Zeigt Optionsmenü an, wenn nicht außer Kraft gesetzt. Wenn außer Kraft gesetzt, legt der Inhalt das Verhalten fest.	Beendet den Player, wenn nicht außer Kraft gesetzt. Wenn außer Kraft gesetzt, legt der Inhalt das Verhalten fest.

Anwendung in Flash testen

Wie schon das Erstellen Ihrer Anwendung ähnelt das Testen Ihrer BREW-Anwendung im Adobe Device Central-Emulator den Abläufen beim Testen einer allgemeinen Flash Lite-Anwendung. Beachten Sie, dass sich Testen im Device Central-Emulator von dem Testen unterscheidet, das Sie mit dem Simulator aus dem BREW SDK durchführen. Die Abläufe beim Testen im Device Central-Emulator werden in diesem Abschnitt, die Abläufe beim Testen im BREW-Simulator unter „Flash Lite-Dateien für BREW veröffentlichen“ auf Seite 103 beschrieben.

Da der Device Central-Emulator derzeit keine Gerätepacks für BREW-Geräte bietet, können mit dem Device Central-Emulator nur die allgemeinen Funktionen getestet werden. Mit dem BREW-Simulator testen Sie die Funktionen Ihrer Anwendung in der BREW-Umgebung.

1 Öffnen Sie die Anwendung in Flash.

2 Wählen Sie „Datei“ > „Einstellungen für Veröffentlichungen“. Wählen Sie auf der Registrierkarte „Flash“ in der Liste „Version“ die Option „Flash Lite“) aus und wählen Sie dann in der ActionScript-Versionsliste „ActionScript 2.0“ aus. Klicken Sie auf „OK“, um Ihre Einstellungen zu speichern.

3 Starten Sie den Adobe Device Central-Emulator (Steuerung > Film testen) und klicken Sie auf die Registerkarte „Geräteprofile“.

4 Führen Sie im Bibliotheksfeld einen der folgenden Schritte aus:

a Wenn Sie noch kein Gerätepack für das Zielgerät heruntergeladen haben, wählen Sie Generisch > Flash Lite 2.0 > Generisches Telefon .

b Wenn Sie ein Gerätepack für das Zielgerät heruntergeladen haben (unter www.adobe.com/products/flash/download/device_profiles), klicken Sie auf den Gerätenamen, um ihn zu erweitern. Wählen Sie die entsprechende Gerätegröße und ziehen Sie den Namen des Zielgeräts aus der Liste Verfügbare Geräte in die Liste Gerätegruppen .

5 Wählen Sie „Datei“ > „Zurück zu Flash“ und wählen Sie im Flash-Menü „Steuerung“ > „Film testen“. Falls der BREW Publisher-Assistent angezeigt wird, klicken Sie auf „Abbrechen“, um das Dialogfeld zu schließen. Die Anwendung wird von Flash in Device Central exportiert und im Emulator geöffnet.

6 Testen Sie Ihre Anwendung im Emulator.

Hinweise zur Verwendung des Emulators finden Sie unter „Emulator verwenden“ auf Seite 119.

Wenn das Testen Ihrer Anwendung im Emulator abgeschlossen ist, speichern Sie Ihre Dateien und führen Sie dann die Anweisungen im Abschnitt „Flash Lite-Dateien für BREW veröffentlichen“ auf Seite 103 aus, um Ihre Anwendung für die BREW-Plattform zu veröffentlichen.

Weitere Informationsquellen

Weiterführende Informationen zur Entwicklung und zum Testen von Flash Lite-Anwendungen finden Sie in den folgenden Quellen in der Flash-Hilfe:

- *Erste Schritte mit Flash Lite 2.x und 3.0* (PDF)
- Flash Lite-Inhalte testen (Hilfethemen)

Flash Lite-Dateien für BREW veröffentlichen

Allgemeine Informationen zum Veröffentlichen von Flash Lite-Dateien für BREW

Flash Lite-Anwendungen müssen dem BREW-Standard entsprechen, damit sie auf BREW-konformen Geräten ausgeführt werden können. Der Flash Lite für BREW Publisher-Assistent vereinfacht Entwicklern das Erstellen von Anwendungen für diesen Standard, indem er einige der erforderlichen Schritte automatisiert. Neben dem Erzeugen der Dateien für die Anwendung selbst verwendet der Assistent das Simulator-Tool aus dem BREW SDK, um Dateien zu erzeugen, die Sie zum Testen Ihrer BREW-Anwendung verwenden können.

Bevor Sie den BREW Publisher-Assistenten verwenden, müssen Sie mit einigen grundlegenden Aspekten der Dateitypen, der Dateistruktur und der Anforderungen an BREW-Anwendungsdateien vertraut sein.

Dateitypen

Der eigentliche Zweck von Flash Lite Publisher für BREW ist das Veröffentlichen einer SWF-Standarddatei, aus der dann von einem Postprozessor eine SWF-Paketdatei erstellt wird, die auf einem BREW-Gerät ausgeführt werden kann. In der folgenden Tabelle werden die zum Verwenden von Flash Lite-Inhalten auf einem BREW-Geräte erforderlichen Dateien aufgelistet:

Datei	Dateiinhalt
MIF-Datei	Konfigurationsinformationen
MOD-Datei	Anwendungsdatei für BREW-Gerät
SIG-Datei	Gerätespezifische Datei von QUALCOMM

Die zum Verwenden von Flash Lite-Inhalt mit dem BREW-Simulator erforderlichen Dateien sind:

Datei	Dateiinhalt
MIF-Datei	Konfigurationsinformationen
DLL-Datei	Anwendungsdatei für den BREW-Simulator

Alle oben genannten Dateien müssen den gleichen Dateinamen aufweisen (plus die jeweilige Dateierweiterung: .mif, .mod, .sig oder .dll); darüber hinaus muss der Dateiname auch der Name des Anwendungsordners sein. Die Dateien müssen sich in den richtigen Ordnern auf dem Gerät befinden, damit die Anwendung ordnungsgemäß ausgeführt werden kann. Die Ordnerstruktur hängt von der BREW-Version ab, die von Ihrem Zielgerät verwendet wird. Weitere Informationen finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.

MIF-Dateien

MIF-Dateien sind Konfigurationsdateien, die von BREW-Applets benötigte Daten enthalten. Der BREW Publisher-Assistent generiert die MIF-Dateien im Rahmen des Veröffentlichungsprozesses. Auch wenn die MIF-Dateien automatisch erstellt werden, muss jeder Entwickler mit diesem Format und dessen Anforderungen vertraut sein und sicherstellen, dass die MIF-Dateieinstellungen korrekt sind, bevor Anwendungen für das True BREW Testing vorgelegt werden.

Im folgenden Code sind die vom BREW Publisher-Assistenten ausgegebenen MIF-Standardwerte für Anwendungen und Bildschirmschoner aufgeführt. Falls einer oder mehrere dieser Werte für Ihre Anwendung falsch sind, laden Sie das MIF Editor-Tool von der BREW-Website und bearbeiten die Dateien, bevor Sie Ihre Anwendungen zum Testen vorlegen.

```
<Applet>
- List of Applets defined in this Module: [Class ID in post processor]
- Applet Information:
* Name: [Applet Name in post processor]
* Class ID: [Class ID in post processor]
* Applet Type: [blank]
- Graphics
* Large: [Large Icon in post processor]
* Medium: [Medium Icon in post processor]
* Small: [Small Icon in post processor]
<Extension>
- Exported Classes: [blank]
- Exported MIME Types: [blank]
<Dependencies>
- External Classes Used by this Module: [blank]
- External Classes Directory (Checkbox): [checked]
- Used: [Class IDs generated by post processor]
<Privileges>
File [checked]
Network [checked]
Web Access [checked]
TAPI [checked]
<Access Control>
- ACL type [blank]
- Rights [blank]
- Groups [blank]
<License>
No License (Checkbox): [checked]
<General>
- Author Name: [Author in post processor]
- Copyright String: [Copyright in post processor]
- Module Version: [Version in post processor]
- EFS Restrictions
* Max Files: [blank]
* Max Space: (bytes) [blank]
- String Encoding Format: [ISOLATIN1]
```

```
Default MIF output for Screensavers
<Applet>
- List of Applets defined in this Module: [Class ID in post processor]
- Applet Information:
* Name: [Applet Name in post processor]
* Class ID: [Class ID in post processor]
* Applet Type: [blank]
- Graphics
* Large: [Large Icon in post processor]
* Medium: [Medium Icon in post processor]
* Small: [Small Icon in post processor]
- Notifications, Flags, Settings…
* Flags: Screensaver [checked]
<Extension>
- Exported Classes: [blank]
- Exported MIME Types: [brew/ssaver Class ID]
<Dependencies>
- External Classes Used by this Module: [blank]
- External Classes Directory (Checkbox): [checked]
- Used: [Class IDs generated by post processor]
<Privileges>
All unchecked
<Access Control>
- ACL type [blank]
- Rights [blank]
- Groups [blank]
<License>
No License (Checkbox): [checked]
<General>
- Author Name: [Author in post processor]
- Copyright String: [Copyright in post processor]
- Module Version: [Version in post processor]
- EFS Restrictions
* Max Files: [blank]
* Max Space: (bytes) [blank]
- String Encoding Format: [ISOLATIN1]
```

SIG-Dateien

SIG-Dateien sind gerätespezifische Signaturdateien, die für jedes Gerät erforderlich sind, für das Sie Anwendungen entwickeln. Diese Dateien können Sie von QUALCOMM beziehen; sie sind 90 Tage lang gültig. Nach 90 Tagen ist die Signaturdatei abgelaufen und QUALCOMM muss eine neue Signaturdatei erstellen.

Es ist nur eine SIG-Datei pro Gerät erforderlich. Da jedoch jede Anwendung eine eigene SIG-Datei benötigt, erstellen Sie eine Kopie der SIG-Originaldatei und benennen sie um, so dass der Dateiname der von Ihnen erstellten Anwendung entspricht. Denken Sie daran, dass sich die SIG-Datei im gleichen Ordner wie die MOD-Datei, die Anwendungsdatei des BREW-Geräts, befinden muss.

Sie müssen ein authentifizierter BREW-Entwickler sein, um auf das Dienstprogramm zur Erzeugung von Signaturdateien auf der BREW-Webseite zugreifen zu können.

Hinweis: *SIG-Dateien sind zwar für das Testen von Anwendungen auf Geräten erforderlich, für das Testen mit dem BREW-Simulator werden sie jedoch nicht benötigt.*

SIG-Dateien erhalten

1 Rufen Sie die BREW-Website auf und gehen Sie zu der Seite, die den Link zum webbasierten Generator von Test-Signaturdateien von QUALCOMM enthält.

2 Wählen Sie „BREW Testing Generator“.

3 Geben Sie Ihren BREW-Entwicklernamen und Ihr Kennwort ein.

4 In das Feld ESN geben Sie **0x<ESN Number>** ein; die ESN-Nummer finden Sie im Batteriefach des Geräts, wenn Sie die Batterie anheben.

5 Klicken Sie auf „Generate“ und warten Sie, bis der Link „Download Signature“ angezeigt wird.

6 Laden Sie die Signatur herunter und speichern Sie die Datei; sie ist für alle Testvorgänge auf dem Gerät erforderlich.

BAR-Dateien

Mit einer Windows-Anwendung mit der Bezeichnung „AppLoader“ (diese Anwendung wird von QUALCOMM als Teil der BREW Tools Suite bereitgestellt) und einem Datenkabel können Sie Dateien auf einem BREW-konformen Gerät hinzufügen, löschen und verschieben. Dateien mit den Erweiterungen .mod , .mif oder .bar können nicht von einem BREW-Gerät auf ein anderes Gerät oder ein Speichermedium kopiert werden. Diese Einschränkung wurde von QUALCOMM eingeführt, um sicherzustellen, dass über das BREW Delivery System (BDS) erworbene Dateien nicht an andere, nicht autorisierte Benutzer verteilt werden können.

Damit auch die Entwickler von Flash-Inhalten von dieser Sicherheitsfunktion profitieren, prüft der auf den BREW-Geräten ausgeführte Flash Lite Player auf das Vorhandensein von BAR-Dateien, wenn er versucht, auf eine Datei auf dem lokalen Gerät zu verweisen. Diese Einschränkung gilt für jede ActionScript- oder Shared Libraries-Funktion, die einen Pfad für den Verweis auf eine Datei verwendet (beispielsweise `loadMovie()`, `loadSound()`, `loadVariables()`, `XML.load()` usw.). Um diese Funktion zu verwenden, können Entwickler allen von ihnen erstellten Dateien die Erweiterung .bar zuweisen; der Player kann sie zwar finden und anzeigen, aber niemand kann sie von dem BREW-Gerät kopieren, auf dass sie ursprünglich geladen wurden. Der Flash Lite Player prüft zuerst, ob die Datei vorhanden ist, indem der angegebene lokale Pfad verwendet wird (z. B. loadMovie(*Beispiel*.swf)). Wird die Datei an diesem Speicherort nicht gefunden, prüft der Player denselben Pfad mit der Erweiterung „.bar“ (z. B. loadMovie(*Beispiel*.swf.bar)). Dann wird die Datei geladen.

Gerätedateistrukturen für verschiedene BREW-Versionen

Geräte, die BREW 2.x und BREW 3.x unterstützen, weisen unterschiedliche Dateistrukturen auf. Diese Unterschiede müssen Ihnen bekannt sein, bevor Sie die Dateistruktur Ihrer BREW-Anwendung mit dem BREW Publisher-Assistenten festlegen.

Das Dateisystem auf BREW 2.x-Geräten ist wie folgt strukturiert:

- Alle Anwendungsordner befinden sich in einem root-Ordner mit der Bezeichnung brew/ .
- Anwendungsdateien (einschließlich der SIG-Dateien) sind im Anwendungsordner gespeichert (Unterordner sind nicht zulässig).
- MIF-Dateien sind im root-Ordner brew/ gespeichert.
- Gemeinsam genutzte Medien wie Grafiken und BAR-Dateien sind im Ordner brew/shared abgelegt. Gemeinsam genutzte Medien können auch in dem Ordner gespeichert werden, der die Hauptdateien der Anwendung enthält.

  Beispiel:

  brew/*Anwendungsname*/*Anwendungsname*.mod

  brew/*Anwendungsname*/*Anwendungsname*.sig

brew/*Anwendungsname*.mif

brew/shared/*Medien*.xxx. xxx ist die Erweiterung beliebiger Ressourcendateien, einschließlich Grafik-, Video-, Sound- und BAR-Dateien.

oder

brew/*Anwendungsname*/*Medien*.xxx. xxx ist die Erweiterung beliebiger Ressourcendateien, einschließlich Grafik-, Video-, Sound- und BAR-Dateien.

Das Dateisystem auf BREW 3.x-Geräten ist wie folgt strukturiert:

- Alle Anwendungsordner sind in einem Ordner mit der Bezeichnung brew/mod gespeichert.
- Anwendungsdateien (einschließlich der SIG-Dateien) sind im Anwendungsordner gespeichert (Unterordner sind nicht zulässig).
- MIF-Dateien sind im Ordner brew/mif gespeichert.
- Gemeinsam genutzte Dateien wie Grafiken und BAR-Dateien können im Ordner brew/shared abgelegt werden. Gemeinsam genutzte Medien können auch in dem Ordner gespeichert werden, der die Hauptdateien der Anwendung enthält.

  Beispiel:

  brew/**mod**/*Anwendungsname*/*Anwendungsname*.mod

  brew/**mod**/*Anwendungsname*/*Anwendungsname*.sig

  brew/**mif**/*Anwendungsname*.mif

  brew/**shared**/*Medien*.xxx. xxx ist die Erweiterung beliebiger Ressourcendateien, einschließlich Grafik-, Video-, Sound- und BAR-Dateien.

  oder

  brew/**mod**/*Anwendungsname*/*Medien*.xxx. xxx ist die Erweiterung beliebiger Ressourcendateien, einschließlich Grafik-, Video-, Sound- und BAR-Dateien.

Arbeitsablauf beim Veröffentlichen und Testen von Flash Lite-Dateien für BREW

Der Arbeitsablauf beim Veröffentlichen und Testen von Flash Lite-Dateien für BREW umfasst folgende Schritte:

- In Flash starten Sie den BREW Publisher-Assistenten und folgen den Anweisungen auf dem Bildschirm, um die erforderlichen Einstellungen für Ihre BREW-Anwendung einzurichten und die Dateien zu veröffentlichen.
- Im BREW SDK verwenden Sie den BREW-Simulator, um Ihre Anwendung zu testen.

BREW Publisher-Assistent zum Veröffentlichen der Anwendungsdateien verwenden

1. Wählen Sie in Flash Datei > Einstellungen für Veröffentlichungen .

   Das Dialogfeld „Einstellungen für Veröffentlichungen“ wird angezeigt.

2. Wählen Sie auf der Registrierkarte „Formate“ die Option „Flash (.swf)“, sofern sie noch nicht ausgewählt ist.

   Sie können alle anderen Dateitypen deaktivieren, da zum Erzeugen des BREW-Applets nur die SWF-Datei erforderlich ist.

3 Wählen Sie auf der Registrierkarte „Flash" in der Liste „Version" eine unterstützte Version des Players (Flash Player 7 oder früher oder eine beliebige Version von Flash Lite) aus und wählen Sie dann eine der Versionen in der ActionScript-Versionsliste aus (es werden alle ActionScript-Versionen unterstützt).

4 Wählen Sie in der Liste „Nachbearbeitungssoftware" unten auf der Registrierkarte „Flash" die Option „Flash Lite für BREW" aus.

Hinweis: *Falls die Liste Nachbearbeitungssoftware nicht auf der Registrierkarte Flash angezeigt wird, führen Sie die Anweisungen unter „System für BREW einrichten" auf Seite 99 aus, um sicherzustellen, dass Flash Lite 2.1 für BREW korrekt installiert ist. Die Liste Nachbearbeitungssoftware wird erst nach der Installation der Software angezeigt.*

1 Klicken Sie auf „Einstellungen" rechts neben der Liste „Nachbearbeitungssoftwar2, um den BREW Publisher-Assistenten anzuzeigen.

2 (Erforderlich) Verwenden Sie die Eingabefelder im Fenster „Identify Applet", um eine eindeutige Klassen-ID und einen Namen für Ihr BREW-Applet anzugeben.

In der folgenden Tabelle sind ausführliche Informationen zu diesen Feldern aufgelistet. Nachdem Sie die Klassen-ID und den Namen Ihres Applets eingegeben haben, klicken Sie auf „Next".

Feldname	Feldinhalt
Class ID	In diesem Feld ist standardmäßig ein zufällig erzeugter Wert eingetragen. Ersetzen Sie diesen Wert durch Ihre zertifizierte Klassen-ID, sofern Sie über eine verfügen. Eine Klassen-ID ist ein eindeutiger 32-Bit-Identifikationscode (8-stelliger hexadezimaler Wert), den der BREW-Mechanismus zum Erstellen einer Schnittstelle verwendet. Beim Testen auf einem lokalen System können Sie den zufällig erzeugten Standardwert verwenden (vorausgesetzt, er ist eindeutig). Dennoch muss QUALCOMM eine eindeutige ID für Anwendungen erzeugen, die für die Veröffentlichung vorgesehen sind. Weitere Informationen zum Beziehen der Klassen-ID-Dateien finden Sie auf der BREW-Website.
Applet Name	In diesem Feld ist der root-Teil des Namens der momentan aktiven FLA-Datei eingetragen. Angenommen, der Name der aktiven FLA-Datei lautet BREW.fla, so lautet der Applet-Name BREW. Der Name, den Sie in dieses Feld eingeben, wird im BREW Application Manager auf dem Zielgerät angezeigt. Für Applet-Namen gelten die folgenden Richtlinien: • Applet-Namen dürfen nur Kleinbuchstaben enthalten und müssen mit einem Buchstaben beginnen. • Zahlen sind zulässig, jedoch nicht als erstes Zeichen. • Der Unterstrich ist das einzige zulässige Sonderzeichen, es darf jedoch nicht das erste Zeichen des Namens sein. • Die maximale Länge des Namens hängt vom Gerät ab. Weitere Informationen finden Sie im Abschnitt zu Gerätespezifikationen auf der BREW-Website.

3 (Optional) In das Fenster „Include Applet Information" können Sie Informationen eingeben, die zusammen mit Ihrem BREW-Applet veröffentlicht werden sollen.

Informationen, die Sie hier eingeben, werden in der MIF-Datei Ihres Applets gespeichert. In der folgenden Tabelle sind ausführliche Informationen zu jedem Feld aufgelistet. Nachdem Sie alle erforderlichen Informationen eingegeben haben, klicken Sie auf „Next".

Feldname	Feldinhalt
Author	Geben Sie den Namen des Applet-Autors ein (maximal 23 Zeichen).
Version	Geben Sie die Applet-Version ein (maximal 23 Zeichen).
Copyright	Geben Sie die Copyright-Informationen ein (maximal 23 Zeichen).
Domain URL	Geben Sie die vollständig qualifizierte Domäne für Ihr Applet ein. Schließen Sie die Eingabe mit einem Schrägstrich ab (Beispiel: http://www.example.com/). Die Domäne des Applets legt fest, auf welche Informationen das Applet zur Laufzeit zugreifen kann. In früheren Versionen von Flash Player konnten SWF-Dateien aus ähnlichen Domänen (beispielsweise www.exampleapps.com und www.examplegames.com) frei miteinander und mit anderen Dokumenten kommunizieren.In Flash Player 7 und höher ist eine Kommunikation nur dann möglich, wenn die Domäne mit den Daten, auf die zugegriffen werden soll, exakt mit der Domäne des Datenproviders übereinstimmt. Weitere Informationen zu Domänen und zur Sicherheit finden Sie in den Themen der Flash-Dokumentation zur Sicherheit, zur domänenübergreifenden Sicherheit und zum Zulassen des Datenzugriffs zwischen domänenübergreifenden SWF-Dateien.
Application Type (Application oder Screen Saver)	Gibt an, ob es sich bei dem von Ihnen erstellten BREW-Applet um eine Anwendung oder um einen Bildschirmschoner handelt.Informationen darüber, wie sich Bildschirmschoner von Anwendungen unterscheiden, finden Sie auf der BREW-Website. Rufen Sie den Bereich Developer FAQs > BREW Tools > MIF Settings > Screensaver auf.

4 (Optional zum Testen Ihrer Anwendung, aber obligatorisch für Anwendungen, die zum True BREW Testing vorgelegt werden.) Geben Sie im Fenster „Applet Icons“ den Namen und Speicherort von mindestens einem Symbol für Ihr Applet ein (oder wählen Sie den Dateipfad aus).

   Folgende Dateitypen werden unterstützt: JPEG, BMP, PNG und BCI (BREW Compressed Image). Diese Symbole werden zusammen mit dem Namen des Applets im BREW Application Manager auf dem Zielgerät angezeigt. Nachdem Sie Ihre Symbole angegeben haben, klicken Sie auf „Next“.

Feldname	Feldinhalt
Small Icon	Grafikdatei für ein kleines Symbol (zur Anzeige auf dem Gerät). Maximale Größe: 16 x 16.
Medium Icon	Grafikdatei für ein mittelgroßes Symbol (zur Anzeige auf dem Gerät). Maximale Größe: 26 x 26.
Large Icon	Grafikdatei für ein großes Symbol (zur Anzeige auf dem Verteilungsserver auf einigen Geräten). Maximale Größe: 65 x 42.

5 Geben Sie im Fenster „Define Output Settings“ die Speicherorte an, an denen Dateien für den BREW-Simulator und das Zielgerät veröffentlicht werden sollen.

   Diese Felder enthalten Standardwerte, die vom Speicherort Ihrer SWF-Datei zusammen mit dem Applet-Namen und in den Ordnernamen (Simulator und Gerät) abgeleitet werden. Diese Angaben legen fest, welche Dateien zum Testen und welche Dateien zum Hochladen auf das Gerät verwendet werden sollen. Sie können diese Standardwerte akzeptieren oder andere Speicherorte für die Ausgabe festlegen. Klicken Sie abschließend auf „Next“.

Feldname	Feldinhalt
For Simulator • MIF Folder • Applet Folder	Enthält die Dateien, die Sie zum Testen Ihres Applets mithilfe des Simulator-Tools im BREW SDK benötigen. Weitere Informationen zu den vorgegebenen Dateispeicherorten und der Ordnerstruktur finden Sie in der Dokumentation zum BREW-Simulator auf der BREW-Website.
For Device	Enthält die Dateien, die Sie an das BREW-Gerät hochladen. Weitere Informationen, wie sich die Ordnerstrukturen je nach der vom Zielgerät unterstützten BREW-Version unterscheiden, finden Sie unter „Dateien an BREW-Geräte hochladen“ auf Seite 112.

6 Im Fenster „Summary of Your Selections“ finden Sie eine Übersicht der von Ihnen gewählten Applet-Einstellungen.

Um eine dieser Einstellungen zu ändern, klicken Sie auf die Schaltfläche „Back“, bis das entsprechende Fenster erneut angezeigt wird. Nehmen Sie dann die Änderung vor. Wenn alle Einstellungen korrekt sind, aktivieren oder deaktivieren Sie das Kontrollkästchen am unteren Rand des Fensters, um festzulegen, ob der Assistent beim Veröffentlichen erneut angezeigt werden soll. Klicken Sie anschließend auf "Fertig stellen", um Ihre Einstellungen zu speichern.

Feldname	Feldinhalt
Summary Of Your Selections	Zeigt eine Liste mit allen von Ihnen gewählten Einstellungen an.
Do Not Show This Wizard On Publish	Legt fest, ob der Assistent erneut angezeigt wird, wenn Sie die Applet-Dateien veröffentlichen. Andernfalls wird der Veröffentlichungsprozess ausgelöst, ohne dass der Assistent erneut aufgerufen wird. Unabhängig von dieser Einstellung können Sie jederzeit auf den Assistenten zugreifen, indem Sie das Dialogfeld „Einstellungen für Veröffentlichungen“ aufrufen und auf die Schaltfläche „Einstellungen“ rechts neben der Liste „Nachbearbeitungssoftware“ klicken.

7 Klicken Sie auf OK , um Ihre Einstellungen zu speichern und wählen Sie dann noch einmal Datei > Einstellungen für Veröffentlichungen . Das Dialogfeld „Einstellungen für Veröffentlichungen“ wird angezeigt.

Hinweis: *Dieser Schritt ist nur dann erforderlich, wenn Sie „Flash Lite für BREW“ das erste Mal für eine bestimmte Datei in der Liste „Nachbearbeitungssoftware“ auswählen. Danach können Sie veröffentlichen, ohne zunächst auf „OK“ zu klicken.*

8 Klicken Sie auf „Veröffentlichen“, um Ihre Dateien an den Speicherorten zu veröffentlichen, die Sie im Assistenten angegeben haben.

Anwendung im BREW-Simulator testen

Bevor Sie Ihre Anwendung an das Gerät hochladen, sollten Sie es mit dem Simulator-Tool im BREW SDK testen. Weitere Informationen zu diesem Tool finden Sie in der Dokumentation zum BREW-Simulator auf der BREW-Website.

1 Wechseln Sie in den Simulator-Ordner für das Applet, das Sie mithilfe des BREW Publisher-Assistenten erstellen.

2 Öffnen Sie den Simulator-Ordner und kopieren Sie den gesamten Inhalt in den BREW SDK-Ordner „Examples“.

Hinweis: *Wenn Sie den BREW SDK-Ordner „Examples“ bereits in Schritt 9 des vorherigen Verfahrens als Speicherort für die Simulator-Ausgabedateien angegeben haben, befinden sich diese Dateien bereits im richtigen Ordner und müssen nicht mehr kopiert werden.*

1 Wählen Sie „Start“ > „Programme“ > „BREW SDK v<*Ihre Version*>“ und wählen Sie den BREW-Simulator, um die Anwendung zu öffnen.

2 Im Simulator wählen Sie File > Load Device und dann das Gerätepack für das Handset, für das Sie Anwendungen entwickeln.

3 Wählen Sie auf der Registrierkarte „Properties“ im Simulator den Ordner Ihrer Anwendung als das Applet Directory.

4 Testen Sie Ihre Anwendung mithilfe der Tools und Funktionen im Simulator.

Gerätepacks verwenden

Wenn Sie authentifizierter BREW-Entwickler sind, können Sie die gerätespezifischen Gerätepacks für Ihre Zielgeräte aus dem Developer Resources-Bereich auf der BREW-Website herunterladen. Das Gerätepack für den Simulator weicht eventuell vom tatsächlichen Gerät ab. Beispielsweise zeigt das Gerätepack für das Samsung SCH-A950 eine BACK-Taste, die auf dem tatsächlichen Gerät mit CLR bezeichnet ist. Darüber hinaus unterstützt das Gerätepack für das LG VX9800 keine Eingabe über die Buchstabentasten der Tastatur; vom tatsächlichen Gerät wird dies jedoch unterstützt. Informationen zu bestimmten Gerätepacks, Aktualisierungen oder bekannten Problemen erhalten Sie von QUALCOMM.

Auch wenn Sie kein authentifizierter BREW-Entwickler sind, können Sie mit den im BREW SDK enthaltenen Standard-Gerätepacks arbeiten. Da diese Standard-Gerätepacks jedoch über weniger Speicherkapazität als die gerätespezifischen Gerätepacks verfügen, müssen Sie eventuell das Gerätepack mit dem größten Arbeitsspeicher (zum Zeitpunkt der Drucklegung Gerätepack 3) verwenden, oder den Arbeitsspeicher auf etwa 10.000.000 Byte erhöhen, um Leistungsprobleme und Fehlermeldungen zu vermeiden.

Arbeitsspeicher für eine einzelne Simulator-Sitzung vergrößern

1 Öffnen Sie den Simulator und wählen Sie die Registrierkarte „Device Details“.

2 Führen Sie einen Bildlauf bis zum Bereich Memory durch.

3 Erhöhen Sie den Wert für „Heap Size (bytes)“. In einigen Simulator-Versionen wird dieser Wert als „RAM Size“ bezeichnet.

Arbeitsspeicher permanent für alle Simulator-Sitzungen vergrößern

1 Öffnen Sie die DSD-Datei für das gewünschte Gerätepack in einem Texteditor (Gerätepack*x*.dsd; *x* ist die Gerätepacknummer.)

2 Suchen Sie die entsprechende Einstellung und erhöhen Sie den Wert für Text:

```
<String Id="24" Name="IDS_DD_HW_RAM_SIZE">
        <Text>1048576</Text>
```

3 Speichern und schließen Sie die Datei.

Auch wenn der Speicherplatz für den Ordner „Examples“ und dessen Unterordner nicht ausreicht, um Ihre Anwendung fehlerfrei auszuführen, können Probleme bei der Arbeit mit dem Simulator auftreten. In diesem Fall können Sie entweder nicht benötigte Dateien aus diesen Ordnern löschen oder den Speicherplatz vergrößern, der diesen Dateien zugewiesen ist.

Permanentes Vergrößern des Speicherplatzes für den Ordner „Examples" für alle Simulator-Sitzungen

1 Öffnen Sie die Konfigurationsdatei in einem Texteditor und ändern Sie den entsprechenden Wert:

- Für das Samsung SCH-A950: Die Konfigurationsdatei lautet SCH-A950.qsc, und der zu ändernde Wert wird unten in Fettschrift angezeigt:

```
FS_LIMITS_PER_MODULE 65535 15204352
```

- Für das LG9800: Die Konfigurationsdatei lautet LG-VX9800.dsd , und der zu ändernde Wert wird unten in Fettschrift angezeigt:

```
<String Id="20" Name="IDS_DD_HW_EFS_SIZE">
<Text>47000000</Text>
<Comment>50MB</Comment>
</String>
```

2 Speichern und schließen Sie die Datei.

Weitere Informationsquellen

Weiterführende Informationen zum BREW-Simulator finden Sie in folgenden Quellen:

Dokumentname und -Typ	Speicherort
Starting with BREW, „About the BREW Tools Suite" (PDF)	Rufen Sie die BREW-Website auf und wählen Sie dann Developer Home > Application Development Documentation .
BREW-Simulator (Hilfe)	Öffnen Sie dem Simulator und klicken Sie in der Menüleiste auf „Help", um die Onlinehilfe anzuzeigen.

Dateien an BREW-Geräte hochladen

Allgemeine Informationen zum Hochladen von Flash Lite-–Anwendungen für BREW

Nachdem Sie Ihre Flash Lite-Anwendung erstellt und mit dem Flash Lite für BREW Publisher-Assistenten BREW-konforme Dateien erzeugt haben, können Sie sie an ein unterstütztes BREW-Gerät hochladen und darauf testen.

Im Idealfall haben Sie bereits während des Designprozesses speziell für ein bestimmtes Gerät entwickelt und Ihre Anwendung mit dem Adobe Device Central-Emulator und dem BREW-Simulator getestet. Dennoch ist es wichtig, Ihre Anwendung sorgfältig auf dem Zielgerät zu testen, um sicherzustellen, dass alle Funktionen Ihren Vorstellungen entsprechend ausgeführt werden.

Bevor Sie Ihre Anwendung hochladen können, benötigen Sie:

- Ein Gerät, das Flash Lite für BREW unterstützt (zum Zeitpunkt der Drucklegung entweder ein Samsung SCH-A950 oder ein LG VX9800).
- Ein serielles oder ein USB-Kabel, um die Anwendung von Ihrem Computer an das Gerät hochzuladen (in der Regel vom Gerätehersteller erhältlich).
- Das Tool AppLoader (verfügbar für authentifizierte BREW-Entwickler als Teil der BREW Tools Suite auf der BREW-Website).
- Anwendungsdateien in den erforderlichen Dateiformaten und -strukturen (erstellt mithilfe des Flash Lite für BREW Publisher-Assistenten).

Folgende Anwendungsdateien sind erforderlich:

- *Anwendungsname*.**SIG** (verfügbar für authentifizierte Entwickler von QUALCOMMs webbasiertem Testsignatur-Generator)
- *Anwendungsname*.**MOD** (erzeugt vom Flash Lite Publisher für BREW-Assistenten)
- *Anwendungsname*.**MIF** (erzeugt vom Flash Lite Publisher für BREW-Assistenten)
- Alle für die Anwendung erforderlichen Sound-, Video-, Grafik- und BAR-Dateien

Vollständige Informationen zu den erforderlichen Dateiformaten und -strukturen für Ihr Zielgerät finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.

Arbeitsablauf beim Hochladen von Anwendungen

Bei diesem Arbeitsablauf wird davon ausgegangen, dass Sie die erforderlichen USB-Treiber für Ihr Gerät bereits auf dem Computer installiert sind. Weitere Informationen finden Sie unter „System für BREW einrichten“ auf Seite 99.

Führen Sie die folgenden Schritte aus, um Flash Lite-Anwendungen an BREW-Geräte hochzuladen und die Anwendungen auf dem Handset zu testen:

1. Schließen Sie das Handgerät an den Computer mit einem Kabel an.
2. Das Tool AppLoader auf dem Computer starten.
   a. Anwendungsdateien an die entsprechenden Speicherorte für das Gerät kopieren. Informationen finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.
   b. Falls erforderlich, die Flash Lite Extension- und Player-Dateien auf das Gerät kopieren (nur beim ersten Mal erforderlich).
   c. Gerät neu starten.
3. Anwendung auf dem Gerät testen.

Diese Aufgaben werden weiter unten in diesem Abschnitt ausführlich beschrieben.

Flash Lite Extension-Dateien an das Gerät hochladen (nur beim ersten Mal erforderlich)

Bevor Sie Ihre Flash Lite für BREW-Anwendungen auf einem bestimmten Gerät testen können, müssen Sie die Flash Lite Extension-Dateien auf das Gerät kopieren. Dieser Schritt muss jedoch nur einmal ausgeführt werden, wenn Sie das Gerät für Testzwecke einrichten. Endbenutzer Ihrer Anwendung erhalten diese Dateien als Teil des Downloads, wenn sie die vollständige Anwendung vom BDS oder ADS herunterladen; Entwickler müssen diese Dateien jedoch manuell kopieren. Die Dateien sind im Folgenden aufgeführt. Führen Sie die Anweisungen weiter unten in diesem Abschnitt aus, um die Dateien mithilfe des Tools AppLoader an Ihr Gerät hochzuladen. Beachten Sie, dass diese Dateien bei BREW 2.x- und 3.x-Geräten an verschiedene Speicherorte kopiert werden müssen. Weitere Informationen finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.

Extension-Datei	Quelle
flashlite_2_1.sig	Wird von Entwicklern mit einem Tool erzeugt, das von der BREW-Website heruntergeladen werden kann.
flashlite_2_1.mif	Kann von der BREW-Website heruntergeladen werden.
flashlite_2_1.mod	Kann von der BREW-Website heruntergeladen werden.

Extension-Datei	Quelle
brewsaplayer.sig	Wird von Entwicklern mit einem Tool erzeugt, das von der BREW-Website heruntergeladen werden kann.
brewsaplayer.mif	Kann von der BREW-Website heruntergeladen werden.
brewsaplayer.mod	Kann von der BREW-Website heruntergeladen werden.

Tipps und bewährte Methoden zum Tool AppLoader

Im Folgenden finden Sie eine Liste mit Tipps zur Verwendung des Tools AppLoader:

- Wenn Sie einen Ordner auf dem Gerät erstellen, verwenden Sie keinen umgekehrten Schrägstrich (\).
- Führen Sie keine weiteren Anwendungen auf dem Gerät aus, während Sie Dateien kopieren.
- Starten Sie das Gerät neu, nachdem Sie Änderungen mit AppLoader vorgenommen haben.
- Vermeiden Sie das Überschreiben von Dateien in AppLoader; löschen Sie stattdessen die alten Versionen der Dateien auf dem Gerät und ersetzen Sie diese durch die neuen Dateien.
- Laden Sie keine Dateien oder Ordner hoch, deren Namen lediglich aus Zahlen bestehen. Andernfalls können Sie diese Dateien und Ordner auf einigen Geräten nicht mehr löschen.
- Beginnen Sie einen Datei- oder Ordnernamen nicht mit der Zeichenfolge „shared“. „Shared“ ist ein reserviertes Wort in BREW; Sie können keine Dateien oder Ordner hochladen, die mit dieser Zeichenfolge beginnen.

Anwendungen an ein BREW 2.x-Gerät hochladen

In den folgenden Anweisungen wird beschrieben, wie Sie eine Flash Lite-Anwendung für BREW an ein BREW-Version 2.x-konformes Gerät hochladen, z. B. das Samsung SCH-A950.

Bei diesem Arbeitsablauf wird davon ausgegangen, dass Sie die erforderlichen USB-Treiber für Ihr Gerät bereits auf Ihrem Computer installiert haben. Weitere Informationen finden Sie unter „System für BREW einrichten“ auf Seite 99.

Bei diesem Arbeitsablauf wird ebenso davon ausgegangen, dass Sie die erforderlichen Dateien für Flash Lite Extension und Flash Lite Player bereits an das Gerät hochladen haben. Dieser Schritt muss jedoch nur einmal ausgeführt werden, wenn Sie das Gerät das erste Mal verwenden. Weitere Informationen finden Sie unter „Flash Lite Extension-Dateien an das Gerät hochladen (nur beim ersten Mal erforderlich)“ auf Seite 113.

1. Schließen Sie das Handset mit dem Datenkabel aus dem Lieferumfang an einem COM-Port des Computers an, auf dem die BREW-Anwendungsdateien gespeichert sind.
2. Wechseln Sie zu dem Ordner, in dem die Anwendungsdateien abgelegt sind und prüfen Sie, ob die Verzeichnisstruktur für das Zielgerät geeignet ist.

   Vollständige Informationen zur Verzeichnisstruktur für BREW 2.x-Geräte finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.
3. Wählen Sie Start > Programme > BREW Tools Suite <*neueste Version* > > BREW Apploader , um das AppLoader-Tool zu starten. Das AppLoader-Tool zeigt ein Verbindungsfenster an.
4. Wählen Sie die Nummer des COM-Anschlusses, an dem Ihr Gerät angeschlossen ist (wenn Sie nicht sicher sind, welchen Anschluss Sie wählen müssen, lesen Sie den unten stehenden Hinweis), wählen Sie „2.x „ als BREW-Version für Ihr Gerät und klicken Sie dann auf „OK“, um eine Verbindung mit dem Gerät herzustellen.

Hinweis: *Angaben zu dem COM-Port, an den Ihr Gerät angeschlossen ist, finden Sie im Windows-Geräte-Manager unter den Eigenschaften des Ports, der dem installierten Gerät entspricht. Wenn Sie immer das gleiche Gerätemodell an den gleichen COM-Port anschließen, ändert sich die Nummer nicht. Sie ändert sich, wenn Sie ein anderes Gerätemodell anschließen. Prüfen Sie die Nummer des COM-Ports daher jedes Mal, wenn Sie ein neues Gerät anschließen.*

Nachdem die Verbindung hergestellt ist, zeigt das AppLoader-Tool das BREW-Dateisystem auf dem Gerät an.

1. Führen Sie einen der folgenden Schritte aus, um Ihre Anwendungsdateien (MOD-, SIG- und Ressourcendateien) an das Gerät hochzuladen:
   - Wählen Sie File > Directory > New Directory , um einen Ordner mit dem gleichen Namen wie Ihre Anwendung zu erstellen und kopieren Sie Ihre Anwendungsdateien dann in diesen Ordner. Einzelheiten finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.
   - Ziehen Sie den Anwendungsordner im Windows Explorer in das AppLoader-Fenster.
2. Gerät neu starten.

   Vor der Ausführung Ihrer Anwendung muss das Gerät neu gestartet werden. Führen Sie einen der folgenden Schritte aus, um das Gerät neu zu starten:
   - Wählen Sie Device > Reset im Menü BREW AppLoader aus.
   - Drücken und halten Sie die End-Taste auf dem Handset.
3. Navigieren Sie zu Get it Now > Get Going und wählen Sie den Namen Ihrer Anwendung aus, um sie in der BREW-Umgebung zu starten.

Anwendungen an ein BREW 3.x-Gerät hochladen

In den folgenden Anweisungen wird beschrieben, wie Sie eine Flash Lite-Anwendung für BREW an ein BREW-Version 3.x-konformes Gerät hochladen, z. B. das LG VX9800.

Bei diesem Arbeitsablauf wird davon ausgegangen, dass Sie die erforderlichen USB-Treiber für Ihr Gerät bereits auf Ihrem Computer installiert haben. Weitere Informationen finden Sie unter „System für BREW einrichten“ auf Seite 99.

Bei diesem Arbeitsablauf wird ebenso davon ausgegangen, dass Sie die erforderlichen Dateien für Flash Lite Extension und Flash Lite Player bereits an das Gerät hochladen haben. Dieser Schritt muss jedoch nur einmal ausgeführt werden, wenn Sie das Gerät das erste Mal verwenden. Weitere Informationen finden Sie unter „Flash Lite Extension-Dateien an das Gerät hochladen (nur beim ersten Mal erforderlich)“ auf Seite 113.

1. Schließen Sie das Handset mit dem Datenkabel aus dem Lieferumfang an einem COM-Port des Computers an, auf dem die BREW-Anwendungsdateien gespeichert sind.
2. Wechseln Sie zu dem Ordner, in dem die Anwendungsdateien abgelegt sind und prüfen Sie, ob die Verzeichnisstruktur für das Zielgerät geeignet ist.

   Vollständige Informationen zur Verzeichnisstruktur für BREW 3.x-Geräte finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.
3. Wählen Sie Start > Programme > BREW Tools Suite <*neueste Version* > > BREW Apploader , um das AppLoader-Tool zu starten.
4. Das AppLoader-Tool zeigt ein Verbindungsfenster an.

   Wählen Sie die Nummer des COM-Anschlusses, an dem Ihr Gerät angeschlossen ist, wählen Sie „3.x“ als BREW-Version für Ihr Gerät und klicken Sie dann auf „OK“, um eine Verbindung mit dem Gerät herzustellen.

Hinweis: *Angaben zu dem COM-Port, an den Ihr Gerät angeschlossen ist, finden Sie im Windows-Geräte-Manager unter den Eigenschaften des Ports, der dem installierten Gerät entspricht. Wenn Sie immer das gleiche Gerätemodell an den gleichen COM-Port anschließen, ändert sich die Nummer nicht. Sie ändert sich, wenn Sie ein anderes Gerätemodell anschließen. Prüfen Sie die Nummer des COM-Ports daher jedes Mal, wenn Sie ein neues Gerät anschließen.*

Nachdem die Verbindung hergestellt ist, zeigt das AppLoader-Tool das BREW-Dateisystem auf dem Gerät an.

1 Führen Sie einen der folgenden Schritte aus, um Ihre Anwendungsdateien (MOD-, SIG- und Ressourcendateien) an das Gerät hochzuladen:

   - Wählen Sie File > New > Directory > New Directory , um einen Ordner mit dem gleichen Namen wie Ihre Anwendung zu erstellen und kopieren Sie die Anwendungsdateien dann in diesen Ordner. Einzelheiten finden Sie unter „Gerätedateistrukturen für verschiedene BREW-Versionen“ auf Seite 106.
   - Ziehen Sie den Anwendungsordner im Windows Explorer in das AppLoader-Fenster.

2 Führen Sie einen der folgenden Schritte aus, um das Gerät neu zu starten:

   - Wählen Sie Device > Reset im Menü BREW AppLoader aus.
   - Drücken und halten Sie die End-Taste auf dem Handset.

3 Navigieren Sie zu Get it Now > Get Going und wählen Sie den Namen Ihrer Anwendung aus, um sie in der BREW-Umgebung zu starten.

   Auf einigen Geräten wird Ihre Anwendung nicht im Standardmenü „Get Going“ angezeigt. Führen Sie die folgenden Schritte aus, um Ihre Anwendungen auch auf diesen Geräten anzuzeigen:

   a Drücken Sie im Verizon-Startfenster die Option „Select/OK“, um auf das Menü zugreifen.

   b Drücken Sie die Null-Taste (0), um das Dienstmenü aufzurufen.

   c Geben Sie das Standard-Kennwort ein. In der Regel ist dies sechs Nullen (000000).

   d Drücken Sie die Neun-Taste (9), um „Get It Now“ aufzurufen.

   e Drücken Sie die g-Taste, um auf das Fenster zuzugreifen, in dem die Flash Lite-Anwendungen angezeigt werden.

Anwendungen auf dem Gerät testen

Bevor Sie Ihre Anwendung auf dem Gerät starten und testen können, müssen Sie eine Testsignaturdatei von QUALCOMM beziehen. Authentifizierte Benutzer können ein webbasiertes Tool mit der Bezeichnung Testsig aus dem BREW Developer-Extranet verwenden, um Testsignaturdateien zu erzeugen. Weiterführende Informationen zum Erzeugen und Verwenden Ihrer Testsignaturdatei finden Sie auf der BREW-Website. Ausführliche Informationen zum Testen Ihrer BREW-Applets, um sicherzustellen, dass sie den TBT-Standards (True BREW Testing) entsprechen, finden Sie im BREW Developer-Extranet.

Weitere Informationsquellen

Weiterführende Informationen zum AppLoader-Tool finden Sie in folgenden Quellen:

Dokumentname und -Typ	**Speicherort**
Starting with BREW, „About the BREW Tools Suite“ (PDF)	Rufen Sie die BREW-Website auf und wählen Sie dann Developer Home > Application Development Documentation .
BREW_Apploader.chm (Hilfe)	C:\Programme\BREW Tools Suite *neueste Version*\BREWApploader (Standard)
BREW Testing-Generator	Rufen Sie die BREW-Website auf und gehen Sie zur SDK-Download-Seite.

Kapitel 9: Flash Lite-Inhalte testen

Zum Lieferumfang von Adobe Flash CS4 Professional gehört ein Adobe Flash Lite-Emulator, der unter Adobe Device Central CS4 verfügbar ist. Mit diesem Emulator können Sie die Wirkung und die Funktionsweise von Anwendungen direkt im Authoring-Tool testen wie auf echten Mobiltelefonen. Wenn die Anwendung im Emulator Ihren Vorstellungen entsprechend aussieht und funktioniert, können Sie sie auf dem eigentlichen Gerät einem Test unterziehen.

Testen – Überblick

Flash Lite-Testfunktionen

Die Flash Lite-Testfunktionen in Flash CS4 Professional sind Teil von Adobe Device Central, was sowohl eine umfangreiche Datenbank mit Geräteprofilen als auch einen Geräte-Emulator umfasst. Device Central kann auch mit vielen anderen Adobe-Produkten verwendet werden, wie etwa Adobe Creative Suite® und Adobe Dreamweaver®.

Mit dem Adobe Device Central-Emulator können Sie sich im Flash-Authoring-Tool eine Vorschau des Flash Lite-Inhalts ansehen. Der Emulator ist so konfiguriert, dass er dem Verhalten sowie dem Erscheinungsbild jedes Zielgeräts entspricht. Über die Device Central-Oberfläche wählen Sie Zielgeräte aus und verwalten diese. Außerdem geben Sie die Flash Lite-Zielinhaltstypen für Ihre Anwendung wie Klingelton-, Browser- oder eigenständige Anwendung an. In Device Central wird mit jeder Kombination aus einem Testgerät und einem Flash Lite-Inhaltstyp eine Gerätekonfiguration definiert, die wiederum die für eine Anwendung verfügbaren Funktionen festlegt, z. B. die unterstützten Audioformate oder die Herstellung von Netzwerkverbindungen. Weitere Informationen über Flash Lite-Inhaltstypen finden Sie unter „Unterstützte Inhaltstypen“ auf Seite 120.

Weitere Informationen einschließlich zur Arbeit mit dem Emulator finden Sie in der Hilfe zu Adobe Device Central.

Vom Emulator nicht unterstützte Testfunktionen

Im Adobe Device Central-Emulator werden nicht alle Funktionen unterstützt, die im Standard-Testfenster der Desktopversion verfügbar sind. Es folgt eine Liste der Funktionen, die im Flash-Desktop-Testfenster verfügbar sind, jedoch nicht im Adobe Device Central-Emulator:

- Die Funktionen „Variablen auflisten“ („Debuggen“ > „Variablen auflisten“) und „Objekte auflisten“ („Debuggen“ > „Objekte auflisten“)
- Der Bandbreiten-Profiler, Streaming-Grafiken und Bild-für-Bild-Grafiken
- Der Befehl Ansicht > Download simulieren
- Der ActionScript-Debugger
- Der Befehl Ansicht > Bildaktualisierungsbereiche anzeigen
- Die Symbolleiste Steuerung („Fenster“ > Symbolleisten > Steuerung)

Inline-Text testen (Flash Lite 2.1). und höher

Derzeit können Sie die Inline-Textfunktion nicht im Adobe Device Central-Emulator testen. Diese Funktion muss auf einem Gerät getestet werden. Wenn Sie im Emulator testen, müssen Sie die Inhalte der Eingabetextfelder mithilfe eines modalen Dialogfelds bearbeiten, das über dem Flash Lite-Inhalt angezeigt wird (d. h. der Emulator arbeitet auf die gleiche Weise wie für Flash Lite 1.x und Flash Lite 2.0). Weitere Informationen zum modalen Dialogfeld finden Sie unter „Eingabetextfelder verwenden“ auf Seite 55.

Emulator verwenden

Adobe Device Central aufrufen

Um den Emulator zu starten, wählen Sie im Flash-Authoring-Tool die Option Steuerung > „Film testen“, genau wie Sie eine Vorschau von Flash-Desktop-Inhalten anzeigen. (Das Adobe Device Central-Testfenster weist jedoch ein unterschiedliches Erscheinungsbild und andere Funktionen als das Testfenster für Flash-Desktop-Inhalte auf.) Wählen Sie Steuerung > „Film testen“, oder drücken Sie <Strg>+<Eingabe>, um Device Central zu starten. Anschließend wird anhand einer Fortschrittsleiste angezeigt, dass die SWF-Datei an Device Central exportiert wird. Im Anschluss an den Export wird Device Central gestartet, wobei der Fokus auf dem Emulator liegt, und die SWF-Datei wird geladen.

Die Gerätesatzliste in Device Central zeigt alle Zielgeräte an, die Flash mit Ihrer Anwendung gespeichert hat. In der Standardeinstellung ist das erste Gerät in der Emulation ausgewählt. Bei dem ausgewählten Inhaltstyp handelt es sich den Inhaltstyp, der zusammen mit der Anwendung bei ihrer Erstellung gespeichert wurde (weitere Informationen zu den Inhaltstypen von Flash Lite finden Sie „Unterstützte Inhaltstypen“ auf Seite 120).

Sie können die Anwendung mit einem anderen Inhaltstyp und anderen Geräten testen. Wenn Sie den Inhaltstyp ändern und ein Gerät aus der Geräteliste entfernen oder ein Gerät zur Liste hinzufügen, werden die Geräteeinstellungen in Flash automatisch geändert.

Testen Sie, wie der Inhalt auf einem anderen Gerät angezeigt wird, indem Sie in der oberen oder unteren Liste auf ein anderes Gerät doppelklicken. Neben dem Gerät, das getestet wird, wird ein sich drehendes Symbol angezeigt und der Emulator zeigt Ihre Anwendung bei der Ausführung auf dem ausgewählten Gerät.

Debugging-Optionen für den Emulator festlegen

Der Adobe Device Central-Emulator kann während der Wiedergabe des Inhalts Debugging-Meldungen an das Flash-Bedienfeld „Ausgabe“ übermitteln. Zudem zeigt der Emulator die Meldungen in einer Popup-Version dieses Bedienfelds an.

Der Emulator zeigt folgende Informationstypen an:

Trace-Meldungen , die von einem `trace()`-Funktionsaufruf in der Flash Lite-Anwendung generiert werden. Weitere Informationen zur Verwendung von `trace()` finden Sie unter `trace()` im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.

Information **Meldungen**, die allgemeine Hinweise zum ausgewählten Testgerät, zur SWF-Dateigröße usw. enthalten. Diese Meldungen werden auf dem Emulator im Bedienfeld „Warnungen“ angezeigt.

Warnung **Meldungen**, die Details zu Problemen mit Flash Lite-Inhalten enthalten, die möglicherweise die Wiedergabe beeinträchtigen.

Sie können den vom Emulator erzeugten Informationstyp wie folgt filtern.

1 Wählen Sie „Steuerung“ > „Film testen“. Die Anwendung wird von Flash in Adobe Device Central exportiert und im Emulator angezeigt.

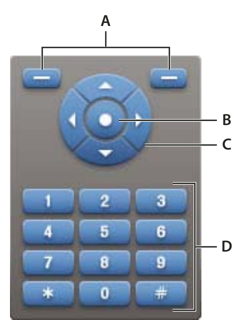
2 Wählen Sie Ansicht> Flash-Ausgabe > In Device Central anzeigen.

- Aktivieren oder deaktivieren Sie die Option „Trace“.
- Aktivieren oder deaktivieren Sie die Option „Informationen“.
- Aktivieren oder deaktivieren Sie die Option „Warnungen“.

Den Emulator bedienen

Sie können die Tastatur des Emulators mithilfe der Computermaus oder mithilfe von Tastenkombinationen bedienen. Auf der Tastatur des Emulators stehen die folgenden Tasten zur Verfügung:

- Zifferntasten (0-9), Sternchen- (*) und Nummerntaste (#)
- Navigationstastatur (Links, Rechts, Unten, Oben, Auswahl)
- Linker und rechter Softkey



A. Softtasten B. Auswahltaste C. Navigationstastatur D. Andere unterstützte Tasten

Sie können entweder mit der Maus direkt auf die Tasten im Emulator klicken oder die folgenden Tastenkombinationen verwenden:

- Die Pfeiltasten auf der Tastatur (<Nach-links >, <Nach-rechts >, <Nach-oben >, <Nach-unten >) entsprechen den Navigationstasten auf der Emulator-Tastatur.
- Die Eingabetaste bzw. der Zeilenschalter entspricht der Auswahltaste des Emulators.
- Die BILD-AUF- und BILD-AB-TASTE entsprechen dem linken und rechten Softkey auf dem Emulator-Keypad.
- Die numerischen Tasten auf der Tastatur entsprechen den numerischen Tasten auf der Emulator-Tastatur.

Weitere Informationen zum Bedienen des Emulators, um eine Anwendung zu testen, finden Sie in der Online-Hilfe zu Adobe Device Central.

Unterstützte Inhaltstypen

Flash Lite wird auf einer Vielzahl von Geräten installiert. Jede Installation unterstützt einen oder mehrere Anwendungsmodi, die auch als *Inhaltstypen* bezeichnet werden. Auf einigen Geräten wird Flash Lite z. B. zur Aktivierung von Bildschirmschonern oder animierten Klingeltönen verwendet. Auf anderen Geräten dient Flash Lite zur Wiedergabe von Flash-Inhalten, die in Webseiten für Mobilgeräte eingebettet sind.

In der nachfolgenden Tabelle werden alle Flash Lite-Inhaltstypen aufgeführt und beschrieben, die zurzeit zur Auswahl stehen. Weitere aktuelle Informationen zur Verfügbarkeit von Flash Lite-Inhaltstypen finden Sie auf der Seite „Flash Enabled Mobile Device" unter www.adobe.com/go/mobile/supported_devices/.

Hinweis: *Zum Zeitpunkt der Drucklegung unterstützt Flash Lite 2.0 nur den Inhaltstyp für eigenständige Player.*

Flash Lite-Inhaltstyp	Beschreibung	Verfügbarkeit
Adressbuch	Ermöglicht es Benutzern, mithilfe von Flash Lite eine SWF-Datei mit einem Eintrag in der Adressbuchanwendung des Geräts zu verknüpfen.	DoCoMo und VodafoneKK (nur Japan)
Alarm	Ermöglicht es Benutzern, mithilfe von Flash Lite eine SWF-Datei auszuwählen, die sie als Alarmton des Geräts verwenden möchten.	KDDI und Vodafone (nur Japan)
Browser	Verwendet Flash Lite, um Flash-Inhalt wiederzugeben, der in Webseiten für Mobilgeräte eingebettet ist und im Webbrowser des Geräts angezeigt wird.	NTT DoCoMo, KDDI und Vodafone (nur Japan)
Anrufprotokoll	Verwendet Flash Lite, um für die einzelnen Einträge im Adressbuch eines Benutzers jeweils ein Bild oder eine Animation sowie den Namen und die Telefonnummer anzuzeigen.	KDDI (nur Casio-Telefone)
Anrufanimation	Verwendet Flash Lite, um eine Animation anzuzeigen, wenn der Benutzer einen Anruf erhält oder tätigt.	NTT DoCoMo und KDDI (nur Japan)
Chaku Flash	Ermöglicht es dem Benutzer mithilfe von Flash Lite, eine SWF-Datei auszuwählen, die als Klingelton für eingehende Anrufe verwendet wird.	KDDI (nur Japan)
Data Box	Verwendet Flash Lite, um Flash-Inhalte in der Data Box-Anwendung des Geräts anzuzeigen, mit deren Hilfe Benutzer Multimedia-Dateien auf dem Gerät verwalten und in einer Vorschau anzeigen können.	NTT DoCoMo, KDDI und Vodafone (nur Japan)
Data Folder	Verwendet Flash Lite, um Flash-Inhalte in der Data Folder-Anwendung des Geräts anzuzeigen, mit deren Hilfe Benutzer Multimedia-Dateien auf dem Gerät verwalten und in einer Vorschau anzeigen können.	KDDI (nur Japan)
Symbolmenü	Ermöglicht es Benutzern, mithilfe von Flash Lite benutzerdefinierte Symbolmenüs für die Launcher-Anwendung des Geräts (vom Inhaltstyp „UI-Launcher" verwendet) auszuwählen.	KDDI (nur Casio-Telefone)
Bildvorschau	Eine Anwendung, die es Benutzern ermöglicht, Multimedia-Dateien (einschließlich SWF-Dateien) auf dem Gerät zu verwalten und in einer Vorschau anzuzeigen.	NTT DoCoMo (nur Japan)

Flash Lite-Inhaltstyp	Beschreibung	Verfügbarkeit
Eingehender Anruf	Verwendet Flash Lite, um eine Animation anzuzeigen, wenn der Benutzer einen Anruf erhält.	NTT DoCoMo, KDDI und Vodafone (nur Japan)
E-Mail-Absender	Verwendet Flash Lite, um eine Animation anzuzeigen, wenn der Benutzer eine E-Mail sendet oder empfängt.	VodafoneKK (nur Japan)
Multimedia	Verwendet Flash Lite, um eine Vorschau von SWF-Dateien (und anderen Multimedia-Formaten) anzuzeigen.	KDDI (nur Japan)
Mein Bild	Eine Anwendung, mit deren Hilfe Benutzer SWF-Dateien und andere Bildformate auf dem Gerät verwalten und in einer Vorschau anzeigen können.	NTT DoCoMo (nur Japan)
OpenEMIRO	Zeigt Flash Lite-Inhalte an, wenn das Gerät aus dem Standby-Modus reaktiviert wird. Vergleichbar mit dem Inhaltstyp „Startbildschirm" auf anderen Geräten.	KDDI (nur Casio-Geräte)
Bildschirmschoner	Verwendet Flash Lite, um den Bildschirmschoner des Geräts anzuzeigen.	KDDI und Vodafone (nur Japan)
SMIL-Player	Verwendet Flash Lite, um eine Vorschau von SWF-Dateien (und anderen Multimedia-Formaten) anzuzeigen.	KDDI (nur Japan)
Eigenständiger Player	Macht Flash Lite als eigenständige Anwendung verfügbar, damit der Benutzer beliebige SWF-Dateien starten und anzeigen kann, die sich auf dem Gerät oder im Posteingang befinden.	Global für bestimmte Symbian Series 60- und UIQ-Geräte verfügbar
Standby-Bildschirm	Verwendet Flash Lite, um den Standby-Bildschirm des Geräts (bzw. Hintergrundbildschirm) anzuzeigen.	NTT DoCoMo und KDDI (nur Japan)
Zweit-Anzeige	Verwendet Flash Lite, um Inhalte auf dem externen oder zweiten Bildschirm anzuzeigen, der auf einigen aufklappbaren Modellen verfügbar ist.	KDDI (nur Japan)
UI-Launcher	Verwendet Flash für den Anwendungs-Launcher des Geräts.	Verwendet Flash Lite, um die Launcher-Anwendung des Geräts anzuzeigen (d. h. die Anwendung, über die der Benutzer andere Anwendungen startet).
Startbildschirm	Verwendet Flash Lite, um beim Starten des Telefons eine Animation anzuzeigen.	NTT DoCoMo (nur Japan)

Flash Lite-spezifische Informationen im Emulator

Der Emulator umfasst Bedienfelder, die Informationen speziell für Ihre Flash Lite-Anwendung bereitstellen. Die Bedienfelder werden auf der rechten Seite des Fensters angezeigt; Sie können sie genau wie in Flash erweitern und minimieren.

Weitere Informationen zu diesen Bedienfeldern und ihrer Verwendung finden Sie in der Online-Hilfe zu Adobe Device Central.

Bildschirmgröße und verfügbare Bühnengröße

Jede Kombination aus einem Zielgerät und einem Flash Lite-Inhaltstyp bestimmt unter anderem, wie groß der verfügbare Bildschirmbereich ist, den eine Flash Lite-Anwendung in Anspruch nehmen kann. Der verfügbare Bühnenbereich füllt den Vollbildschirm des Geräts entweder ganz oder nur zum Teil aus.

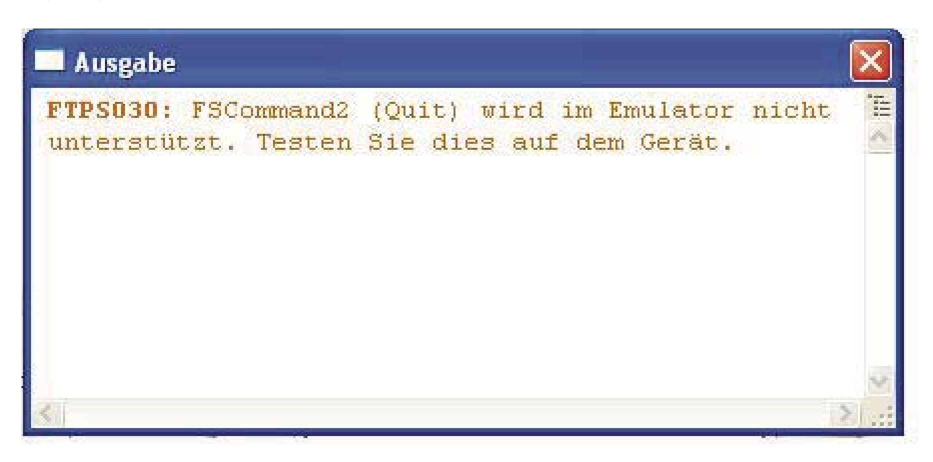
Beispielsweise stimmt die Größe des Bühnenbereichs, der für eine im eigenständigen Player auf einem Nokia Series 60-Gerät im Vollbildmodus ausgeführte SWF-Datei verfügbar ist, mit der Größe des Vollbildschirms des Geräts (176 x 208 Pixel) überein. Auf anderen Geräten (z. B. auf in Japan angebotenen Telefonen) ist der verfügbare Bühnenbereich für eine SWF-Datei, die in einem der speziellen Inhaltstypen (z. B. Adressbuch oder Bildschirmschoner) ausgeführt wird, möglicherweise kleiner als der Vollbildschirm des Geräts. Beispielsweise hat das Fujitsu 700i-Gerät eine Bildschirmgröße von 240 x 320. Für eine SWF-Datei, die in der Adressbuch-Anwendung des Geräts ausgeführt wird, ist aber nur ein Bühnenbereich von 96 x 72 Pixel verfügbar.

Wenn die Abmessungen einer SWF-Datei und die verfügbare Bühnengröße nicht übereinstimmen, skaliert der Flash Lite Player den Inhalt proportional so, dass er in den verfügbaren Bühnenbereich passt. Wenn Sie einen Inhalt im Emulator testen und dabei festgestellt wird, dass sich die Bühnengröße der Anwendung vom verfügbaren Bühnenbereich unterscheidet, wird außerdem eine Warnmeldung angezeigt:

Achten Sie darauf, dass die Bühnengröße Ihres Flash-Dokuments dem verfügbaren Bühnenbereich des ausgewählten Testgeräts und Inhaltstyps entspricht, um Probleme zu vermeiden, die durch eine unerwünschte Skalierung verursacht werden.

Vom Emulator nicht unterstützte Flash Lite-Funktionen

Der Emulator unterstützt nicht alle ActionScript-Befehle und Player-Funktionen, die für Flash Lite-Anwendungen auf einem tatsächlichen Gerät zur Verfügung stehen. Beispielsweise ist es im Emulator nicht möglich, Telefonanrufe oder SMS-Nachrichten zu initiieren. Wenn Sie versuchen, einen Befehl oder eine Funktion zu verwenden, der bzw. die vom Emulator nicht unterstützt wird, wird im Bedienfeld „Ausgabe" eine Meldung wie die in der folgenden Abbildung angezeigt:



Sie müssen Ihre SWF-Datei auf einem tatsächlichen Gerät testen, um sicherzustellen, dass diese Funktionen erwartungsgemäß ausgeführt werden.

Der Adobe Device Central-Emulator bietet keine Unterstützung für die folgenden `fscommand()`- und `fscommand2()`-Befehle:

- `FullScreen`
- `GetFreePlayerMemory`
- `GetTotalPlayerMemory`
- `Launch`
- `Quit`
- `StartVibrate`
- `GetNetworkConnectStatus`
- `GetNetworkRequestStatus`
- `GetNetworkStatus`

Gerätevideo im Emulator wiedergeben

Der Flash Lite Player gibt Videoinhalte in der SWF-Datei mit der Standardvideoanwendung des Geräts wieder, anstatt das Video nativ zu dekodieren. Durch diese Vorgehensweise können Flash Lite-Entwickler beliebige Videoformate verwenden, die auf dem Zielgerät unterstützt werden, z. B. 3GPP, MPEG oder AVI. Weitere Informationen zur Verwendung von Video in Flash Lite finden Sie unter „Mit Videos arbeiten“ auf Seite 79.

Beim Testen im Flash-Authoring-Tool gibt der Adobe Device Central-Emulator Gerätevideo mit QuickTime Player wieder. Die aktuelle Version von QuickTime Player (Zum Zeitpunkt der Drucklegung Version 7) unterstützt die Wiedergabe verschiedener Gerätevideoformate, einschließlich 3GPP und anderen. In der Standardeinstellung unterstützt QuickTime jedoch möglicherweise die Wiedergabe einiger Videoformate nicht, die auf einem tatsächlichen Gerät unterstützt werden. Diese Formate können daher im Adobe Device Central-Emulator nicht wiedergegeben werden. Aus diesem Grund ist es wichtig, Inhalte immer auf einem tatsächlichen Gerät zu testen.

Wenn das Gerätevideo in der Standardeinstellung von QuickTime Player nicht wiedergegeben werden kann, versuchen Sie Folgendes:

- Aktualisieren Sie auf die aktuelle Version von QuickTime Player.
- Installieren Sie einen Video-Codec (kurz für „Codierer/Decodierer“) eines Drittanbieters, der das verwendete Videoformat unterstützt, sofern ein solcher Codec verfügbar ist.

Kapitel 10: Einführung in Flash Lite 2.x und 3.x ActionScript

Macromedia Flash Lite 1.0 und Adobe Flash Lite 1.1 von Adobe, die ersten Versionen von Flash Lite, basieren auf der Macromedia Flash Player 4-Software von Adobe. Macromedia Flash Lite 2.0 und Macromedia Flash Lite 2.1 von Adobe sowie Adobe Flash Lite 3.0 und 3.1 basieren auf Macromedia Flash Player 7 von Adobe, unterscheiden sich hiervon jedoch in den folgenden Punkten:

- Flash Lite unterstützt einige Funktionen nur teilweise.
- Flash Lite enthält einige zusätzliche Funktionen speziell für Mobilgeräte.

In diesem Dokument werden die Unterschiede zwischen dem für Flash Lite 2.0, 2.1, 3.0 und 3.1 (gemeinsam als *2.x* und 3.x bezeichnet) unterstützten Adobe ActionScrip-Befehlsumfang und dem für Flash Player 7 beschrieben.

Unterstützte, teilweise unterstützte und nicht unterstützte ActionScript-Klassen und -Sprachelemente

Eingabehilfen

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Accessibility-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
isActive() : Boolean	Ja	Nein	Nein	Nein
updateProperties() : Void	Ja	Nein	Nein	Nein

arguments

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die arguments-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
callee:Object	Ja	Ja	Ja	Ja
caller:Object	Ja	Ja	Ja	Ja
length:Number	Ja	Ja	Ja	Ja

Array

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Array-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
CASEINSENSITIVE:Number	Ja	Ja	Ja	Ja
concat([value:Object]) : Array	Ja	Ja	Ja	Ja
DESCENDING:Number	Ja	Ja	Ja	Ja
join([delimiter:String]) : String	Ja	Ja	Ja	Ja
length:Number	Ja	Ja	Ja	Ja
NUMERIC:Number	Ja	Ja	Ja	Ja
pop() : Object	Ja	Ja	Ja	Ja
push(value:Object) : Number	Ja	Ja	Ja	Ja
RETURNINDEXEDARRAY:Number	Ja	Ja	Ja	Ja
reverse() : Void	Ja	Ja	Ja	Ja
shift() : Object	Ja	Ja	Ja	Ja
slice([startIndex:Number], [endIndex:Number]) : Array	Ja	Ja	Ja	Ja
sort([compareFunction:Object], [options:Number]) : Array	Ja	Ja	Ja	Ja
sortOn(fieldName:Object, [options:Object]) : Array	Ja	Ja	Ja	Ja
splice(startIndex:Number, [deleteCount:Number], [value:Object]) : Array	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja
UNIQUESORT:Number	Ja	Ja	Ja	Ja
unshift(value:Object) : Number	Ja	Ja	Ja	Ja

AsBroadcaster

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die AsBroadcaster-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_listeners:Array [schreibgeschützt]	Ja	Nein	Ja	Ja
addListener(listenerObj:Object) : Boolean	Ja	Nein	Ja	Ja
broadcastMessage(eventName:String) : Void	Ja	Nein	Ja	Ja
initialize(obj:Object) : Void	Ja	Nein	Ja	Ja
removeListener(listenerObj:Object) : Boolean	Ja	Nein	Ja	Ja

BevelFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die BevelFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
angle:Number	Ja	Nein	Nein	Nein
BevelFilter([distance:Number], [angle:Number], [highlightColor:Number], [highlightAlpha:Number], [shadowColor:Number], [shadowAlpha:Number], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [type:String], [knockout:Boolean])	Ja	Nein	Nein	Nein
blurX:Number	Ja	Nein	Nein	Nein
blurY:Number	Ja	Nein	Nein	Nein
clone() : BevelFilter	Ja	Nein	Nein	Nein
distance:Number	Ja	Nein	Nein	Nein
highlightAlpha:Number	Ja	Nein	Nein	Nein
highlightColor:Number	Ja	Nein	Nein	Nein
knockout:Boolean	Ja	Nein	Nein	Nein
quality:Number	Ja	Nein	Nein	Nein
shadowAlpha:Number	Ja	Nein	Nein	Nein
shadowColor:Number	Ja	Nein	Nein	Nein
strength:Number	Ja	Nein	Nein	Nein
type:String	Ja	Nein	Nein	Nein

BitmapData

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die BitmapData-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
applyFilter(sourceBitmap:BitmapData , sourceRect:Rectangle, destPoint:Point, filter:BitmapFilter) : Number	Ja	Nein	Nein	Nein
BitmapData ((width:Number, height:Number, [transparent:Boolean], [fillColor:Number])	Ja	Nein	Nein	Ja
clone() : BitmapData	Ja	Nein	Nein	Ja
colorTransform(rect:Rectangle, colorTransform:ColorTransform) : Void	Ja	Nein	Nein	Ja
compare(otherBitmapData:BitmapData) : Object	Ja	Nein	Nein	Nein

Name der Funktion	**ActionScript 2**	**Flash Lite 2.1**	**Flash Lite 3.0**	**Flash Lite 3.1**
copyChannel(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, sourceChannel:Number, destChannel:Number) : Void	Ja	Nein	Nein	Ja
copyPixels(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, [alphaBitmap:BitmapData], [alphaPoint:Point], [mergeAlpha:Boolean]) : Void	Ja	Nein	Nein	Teilweise
dispose() : Void	Ja	Nein	Nein	Ja
draw(source:Object, [matrix:Matrix], [colorTransform:ColorTransform], [blendMode:Object], [clipRect:Rectangle], [smooth:Boolean]) : Void	Ja	Nein	Nein	Teilweise
fillRect(rect:Rectangle, color:Number) : Void	Ja	Nein	Nein	Ja
floodFill(x:Number, y:Number, color:Number) : Void	Ja	Nein	Nein	Ja
generateFilterRect(sourceRect:Rectangle, filter:BitmapFilter) : Rectangle	Ja	Nein	Nein	Nein
getColorBoundsRect(mask:Number, color:Number, [findColor:Boolean]) : Rectangle	Ja	Nein	Nein	Ja
getPixel(x:Number, y:Number) : Number	Ja	Nein	Nein	Ja
getPixel32(x:Number, y:Number) : Number	Ja	Nein	Nein	Ja
height:Number [schreibgeschützt]	Ja	Nein	Nein	Ja
hitTest(firstPoint:Point, firstAlphaThreshold:Number, secondObject:Object, [secondBitmapPoint:Point], [secondAlphaThreshold:Number]) : Boolean	Ja	Nein	Nein	Ja
loadBitmap(id:String) : BitmapData	Ja	Nein	Nein	Ja
merge(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, redMult:Number, greenMult:Number, blueMult:Number, alphaMult:Number) : Void	Ja	Nein	Nein	Ja
noise(randomSeed:Number, [low:Number], [high:Number], [channelOptions:Number], [grayScale:Boolean]) : Void	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
paletteMap(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, [redArray:Array], [greenArray:Array], [blueArray:Array], [alphaArray:Array]) : Void	Ja	Nein	Nein	Nein
perlinNoise(baseX:Number, baseY:Number, numOctaves:Number, randomSeed:Number, stitch:Boolean, fractalNoise:Boolean, [channelOptions:Number], [grayScale:Boolean], [offsets:Object]) : Void	Ja	Nein	Nein	Nein
pixelDissolve(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, [randomSeed:Number], [numberOfPixels:Number], [fillColor:Number]) : Number	Ja	Nein	Nein	Nein
rectangle:Rectangle [schreibgeschützt]	Ja	Nein	Nein	Ja
scroll(x:Number, y:Number) : Void	Ja	Nein	Nein	Nein
setPixel(x:Number, y:Number, color:Number) : Void	Ja	Nein	Nein	Ja
setPixel32(x:Number, y:Number, color:Number) : Void	Ja	Nein	Nein	Ja
threshold(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, operation:String, threshold:Number, [color:Number], [mask:Number], [copySource:Boolean]) : Number	Ja	Nein	Nein	Nein
transparent:Boolean [schreibgeschützt]	Ja	Nein	Nein	Ja
width:Number [schreibgeschützt]	Ja	Nein	Nein	Ja

BitmapFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die BitmapFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
clone() : BitmapFilter	Ja	Nein	Nein	Nein

BlurFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die BlurFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
BlurFilter([blurX:Number], [blurY:Number], [quality:Number])	Ja	Nein	Nein	Nein
blurX:Number	Ja	Nein	Nein	Nein
blurY:Number	Ja	Nein	Nein	Nein
clone() : BlurFilter	Ja	Nein	Nein	Nein
quality:Number	Ja	Nein	Nein	Nein

Boolean

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Boolean-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
Boolean([value:Object])	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja
valueOf() : Boolean	Ja	Ja	Ja	Ja

Schaltfläche

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Button-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Ja	Ja	Ja	Ja
_focusrect:Boolean	Ja	Ja	Ja	Ja
_height:Number	Ja	Ja	Ja	Ja
_highquality:Number	Veraltet	Veraltet	Veraltet	Veraltet
_name:String	Ja	Ja	Ja	Ja
_parent:MovieClip	Ja	Ja	Ja	Ja
_quality:String	Ja	Ja	Ja	Ja
_rotation:Number	Ja	Ja	Ja	Ja
_soundbuftime:Number	Ja	Ja	Ja	Ja
_target:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_url:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_visible:Boolean	Ja	Ja	Ja	Ja
_width:Number	Ja	Ja	Ja	Ja
_x:Number	Ja	Ja	Ja	Ja
_xmouse:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_xscale:Number	Ja	Ja	Ja	Ja
_y:Number	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_ymouse:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_yscale:Number	Ja	Ja	Ja	Ja
blendMode:Object	Ja	Nein	Nein	Nein
cacheAsBitmap:Boolean	Ja	Nein	Nein	Nein
enabled:Boolean	Ja	Ja	Ja	Ja
filters:Array	Ja	Nein	Nein	Nein
getDepth() : Number	Ja	Ja	Ja	Ja
menu:ContextMenu	Ja	Nein	Nein	Nein
onDragOut = function() {}	Ja	Ja	Ja	Ja
onDragOver = function() {}	Ja	Ja	Ja	Ja
onKeyDown = function() {}	Ja	Ja	Ja	Ja
onKeyUp = function() {}	Ja	Ja	Ja	Ja
onKillFocus = function(newFocus:Object) {}	Ja	Ja	Ja	Ja
onPress = function() {}	Ja	Ja	Ja	Ja
onRelease = function() {}	Ja	Ja	Ja	Ja
onReleaseOutside = function() {}	Ja	Ja	Ja	Ja
onRollOut = function() {}	Ja	Ja	Ja	Ja
onRollOver = function() {}	Ja	Ja	Ja	Ja
onSetFocus = function(oldFocus:Object) {}	Ja	Ja	Ja	Ja
scale9Grid:Rectangle	Ja	Nein	Nein	Nein
tabEnabled:Boolean	Ja	Ja	Ja	Ja
tabIndex:Number	Ja	Ja	Ja	Ja
trackAsMenu:Boolean	Ja	Ja	Ja	Ja
useHandCursor:Boolean	Ja	Nein	Nein	Nein

Camera

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Camera-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
activityLevel:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
bandwidth:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
currentFps:Number [schreibgeschützt]	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
fps:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
get([index:Number]) : Camera	Ja	Nein	Nein	Nein
height:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
index:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
motionLevel:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
motionTimeOut:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
muted:Boolean [schreibgeschützt]	Ja	Nein	Nein	Nein
name:String [schreibgeschützt]	Ja	Nein	Nein	Nein
names:Array [schreibgeschützt]	Ja	Nein	Nein	Nein
onActivity = function(active:Boolean) {}	Ja	Nein	Nein	Nein
onStatus = function(infoObject:Object) {}	Ja	Nein	Nein	Nein
quality:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
setMode([width:Number], [height:Number], [fps:Number], [favorArea:Boolean]) : Void	Ja	Nein	Nein	Nein
setMotionLevel([motionLevel:Number], [timeOut:Number]) : Void	Ja	Nein	Nein	Nein
setQuality([bandwidth:Number], [quality:Number]) : Void	Ja	Nein	Nein	Nein
width:Number [schreibgeschützt]	Ja	Nein	Nein	Nein

capabilities

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die capabilities-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
audioMIMETypes:Array [schreibgeschützt]	Nein	Ja	Ja	Ja
avHardwareDisable:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
has4WayKeyAS:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasAccessibility:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasAudio:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasAudioEncoder:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasCMIDI:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
hasCompoundSound:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasDataLoading:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasEmail:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasEmbeddedVideo:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasIME:Boolean [schreibgeschützt]	Ja	Nein	Nein	Nein
hasMappableSoftKeys:Boolean	Nein	Ja	Ja	Ja
hasMFI:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasMIDI:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasMMS:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasMouse:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasMP3:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasPrinting:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasQWERTYKeyboard:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasScreenBroadcast:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasScreenPlayback:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasSharedObjects:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasSMAF:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasSMS:Number [schreibgeschützt]	Nein	Ja	Ja	Ja
hasStreamingAudio:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasStreamingVideo:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
hasStylus:Boolean [schreibgeschützt]	Nein	Ja	Ja	Ja
hasVideoEncoder:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
isDebugger:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
language:String [schreibgeschützt]	Ja	Nein	Nein	Nein
localFileReadDisable:Boolean [schreibgeschützt]	Ja	Ja	Ja	Ja
manufacturer:String [schreibgeschützt]	Ja	Nein	Nein	Nein
MIMETypes:Array [schreibgeschützt]	Nein	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
os:String [schreibgeschützt]	Ja	Nein	Nein	Nein
pixelAspectRatio:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
playerType:String [schreibgeschützt]	Ja	Nein	Nein	Nein
screenColor:String [schreibgeschützt]	Ja	Nein	Nein	Nein
screenDPI:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
screenOrientation:String [schreibgeschützt]	Nein	Ja	Ja	Ja
screenResolutionX:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
screenResolutionY:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
serverString:String [schreibgeschützt]	Ja	Nein	Nein	Nein
softKeyCount:Number [schreibgeschützt]	Nein	Ja	Ja	Ja
version:String [schreibgeschützt]	Ja	Ja	Ja	Ja
videoMIMETypes:Array [schreibgeschützt]	Nein	Ja	Ja	Ja

color

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Color-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getTransform() : Object	Veraltet	Ja	Veraltet	Veraltet
setRGB(offset:Number) : Void	Veraltet	Ja	Veraltet	Veraltet
setTransform(transformObject:Object) : Void	Veraltet	Ja	Veraltet	Veraltet

ColorMatrixFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ColorMatrixFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
clone() : ColorMatrixFilter	Ja	Nein	Nein	Nein
ColorMatrixFilter(matrix:Array)	Ja	Nein	Nein	Nein
matrix:Array	Ja	Nein	Nein	Nein

ColorTransform

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ColorTransform-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alphaMultiplier:Number	Ja	Nein	Nein	Ja
alphaOffset:Number	Ja	Nein	Nein	Ja
blueMultiplier:Number	Ja	Nein	Nein	Ja
blueOffset:Number	Ja	Nein	Nein	Ja
ColorTransform([redMultiplier:Number], [greenMultiplier:Number], [blueMultiplier:Number], [alphaMultiplier:Number], [redOffset:Number], [greenOffset:Number], [blueOffset:Number], [alphaOffset:Number])	Ja	Nein	Nein	Ja
concat(second:ColorTransform) : Void	Ja	Nein	Nein	Ja
greenMultiplier:Number	Ja	Nein	Nein	Ja
greenOffset:Number	Ja	Nein	Nein	Ja
redMultiplier:Number	Ja	Nein	Nein	Ja
redOffset:Number	Ja	Nein	Nein	Ja
rgb:Number	Ja	Nein	Nein	Ja
toString() : String	Ja	Nein	Nein	Ja

ContextMenu

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ContextMenu-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
builtInItems:Object	Ja	Nein	Nein	Nein
ContextMenu([callbackFunction:Function])	Ja	Nein	Nein	Nein
customItems:Array	Ja	Nein	Nein	Nein
hideBuiltInItems() : Void	Ja	Nein	Nein	Nein
onSelect = function(item:Object, item_menu:Object) {}	Ja	Nein	Nein	Nein

ContextMenuItem

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ContextMenuItem-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
caption:String	Ja	Nein	Nein	Nein
ContextMenuItem(caption:String, callbackFunction:Function, [separatorBefore:Boolean], [enabled:Boolean], [visible:Boolean])	Ja	Nein	Nein	Nein
copy() : ContextMenuItem	Ja	Nein	Nein	Nein
enabled:Boolean	Ja	Nein	Nein	Nein
onSelect = function(obj:Object, menuItem:Object) {}	Ja	Nein	Nein	Nein
separatorBefore:Boolean	Ja	Nein	Nein	Nein
visible:Boolean	Ja	Nein	Nein	Nein

ConvolutionFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ConvolutionFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Ja	Nein	Nein	Nein
bias:Number	Ja	Nein	Nein	Nein
clamp:Boolean	Ja	Nein	Nein	Nein
clone() : ConvolutionFilter	Ja	Nein	Nein	Nein
color:Number	Ja	Nein	Nein	Nein
ConvolutionFilter(matrixX:Number, matrixY:Number, matrix:Array, [divisor:Number], [bias:Number], [preserveAlpha:Boolean], [clamp:Boolean], [color:Number], [alpha:Number])	Ja	Nein	Nein	Nein
divisor:Number	Ja	Nein	Nein	Nein
matrix:Array	Ja	Nein	Nein	Nein
matrixX:Number	Ja	Nein	Nein	Nein
matrixY:Number	Ja	Nein	Nein	Nein
preserveAlpha:Boolean	Ja	Nein	Nein	Nein

CustomActions

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die CustomActions-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
get(name:String) : String	Ja	Nein	Nein	Nein
install(name:String, data:String) : Boolean	Ja	Nein	Nein	Nein
list() : Array	Ja	Nein	Nein	Nein
uninstall(name:String) : Boolean	Ja	Nein	Nein	Nein

Date

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Date-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
Date([yearOrTimevalue:Number], [month:Number], [date:Number], [hour:Number], [minute:Number], [second:Number], [millisecond:Number])	Ja	Ja	Ja	Ja
getDate() : Number	Ja	Ja	Ja	Ja
getDay() : Number	Ja	Ja	Ja	Ja
getFullYear() : Number	Ja	Ja	Ja	Ja
getHours() : Number	Ja	Ja	Ja	Ja
getLocaleLongDate() : String	Nein	Ja	Ja	Ja
getLocaleShortDate() : String	Nein	Ja	Ja	Ja
getLocaleTime() : String	Nein	Ja	Ja	Ja
getMilliseconds() : Number	Ja	Ja	Ja	Ja
getMinutes() : Number	Ja	Ja	Ja	Ja
getMonth() : Number	Ja	Ja	Ja	Ja
getSeconds() : Number	Ja	Ja	Ja	Ja
getTime() : Number	Ja	Ja	Ja	Ja
getTimezoneOffset() : Number	Ja	Ja	Ja	Ja
getUTCDate() : Number	Ja	Ja	Ja	Ja
getUTCDay() : Number	Ja	Ja	Ja	Ja
getUTCFullYear() : Number	Ja	Ja	Ja	Ja
getUTCHours() : Number	Ja	Ja	Ja	Ja
getUTCMilliseconds() : Number	Ja	Ja	Ja	Ja
getUTCMinutes() : Number	Ja	Ja	Ja	Ja
getUTCMonth() : Number	Ja	Ja	Ja	Ja
getUTCSeconds() : Number	Ja	Ja	Ja	Ja
getUTCYear() : Number	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getYear() : Number	Ja	Ja	Ja	Ja
setDate(date:Number) : Number	Ja	Ja	Ja	Ja
setFullYear(year:Number, [month:Number], [date:Number]) : Number	Ja	Ja	Ja	Ja
setHours(hour:Number) : Number	Ja	Ja	Ja	Ja
setMilliseconds(millisecond:Number) : Number	Ja	Ja	Ja	Ja
setMinutes(minute:Number) : Number	Ja	Ja	Ja	Ja
setMonth(month:Number, [date:Number]) : Number	Ja	Ja	Ja	Ja
setSeconds(second:Number) : Number	Ja	Ja	Ja	Ja
setTime(millisecond:Number) : Number	Ja	Ja	Ja	Ja
setUTCDate(date:Number) : Number	Ja	Ja	Ja	Ja
setUTCFullYear(year:Number, [month:Number], [date:Number]) : Number	Ja	Ja	Ja	Ja
setUTCHours(hour:Number, [minute:Number], [second:Number], [millisecond:Number]) : Number	Ja	Ja	Ja	Ja
setUTCMilliseconds(millisecond:Number) : Number	Ja	Ja	Ja	Ja
setUTCMinutes(minute:Number, [second:Number], [millisecond:Number]) : Number	Ja	Ja	Ja	Ja
setUTCMonth(month:Number, [date:Number]) : Number	Ja	Ja	Ja	Ja
setUTCSeconds(second:Number, [millisecond:Number]) : Number	Ja	Ja	Ja	Ja
setYear(year:Number) : Number	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja
UTC(year:Number, month:Number, [date:Number], [hour:Number], [minute:Number], [second:Number], [millisecond:Number]) : Number	Ja	Ja	Ja	Ja
valueOf() : Number	Ja	Ja	Ja	Ja

DisplacementMapFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die DisplacementMapFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Ja	Nein	Nein	Nein
clone() : DisplacementMapFilter	Ja	Nein	Nein	Nein
color:Number	Ja	Nein	Nein	Nein
componentX:Number	Ja	Nein	Nein	Nein
componentY:Number	Ja	Nein	Nein	Nein
DisplacementMapFilter(mapBitmap:BitmapData, mapPoint:Point, componentX:Number, componentY:Number, scaleX:Number, scaleY:Number, [mode:String], [color:Number], [alpha:Number])	Ja	Nein	Nein	Nein
mapBitmap:BitmapData	Ja	Nein	Nein	Nein
mapPoint:Point	Ja	Nein	Nein	Nein
mode:String	Ja	Nein	Nein	Nein
scaleX:Number	Ja	Nein	Nein	Nein
scaleY:Number	Ja	Nein	Nein	Nein

DropShadowFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die DropShadowFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Ja	Nein	Nein	Nein
angle:Number	Ja	Nein	Nein	Nein
blurX:Number	Ja	Nein	Nein	Nein
blurY:Number	Ja	Nein	Nein	Nein
clone() : DropShadowFilter	Ja	Nein	Nein	Nein
color:Number	Ja	Nein	Nein	Nein
distance:Number	Ja	Nein	Nein	Nein
DropShadowFilter([distance:Number], [angle:Number], [color:Number], [alpha:Number], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [inner:Boolean], [knockout:Boolean], [hideObject:Boolean])	Ja	Nein	Nein	Nein
hideObject:Boolean	Ja	Nein	Nein	Nein
inner:Boolean	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
knockout:Boolean	Ja	Nein	Nein	Nein
quality:Number	Ja	Nein	Nein	Nein
strength:Number	Ja	Nein	Nein	Nein

Error

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Error-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
Error([message:String])	Ja	Ja	Ja	Ja
message:String	Ja	Ja	Ja	Ja
name:String	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja

ExtendedKey

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ExtendedKey-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
SOFT1:String	Nein	Ja	Ja	Ja
SOFT10:String	Nein	Ja	Ja	Ja
SOFT11:String	Nein	Ja	Ja	Ja
SOFT12:String	Nein	Ja	Ja	Ja
SOFT2:String	Nein	Ja	Ja	Ja
SOFT3:String	Nein	Ja	Ja	Ja
SOFT4:String	Nein	Ja	Ja	Ja
SOFT5:String	Nein	Ja	Ja	Ja
SOFT6:String	Nein	Ja	Ja	Ja
SOFT7:String	Nein	Ja	Ja	Ja
SOFT8:String	Nein	Ja	Ja	Ja
SOFT9:String	Nein	Ja	Ja	Ja

ExternalInterface

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die ExternalInterface-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addCallback(methodName:String, instance:Object, method:Function) : Boolean	Ja	Nein	Ja	Ja
available:Boolean [schreibgeschützt]	Ja	Nein	Ja	Ja
call(methodName:String, [parameter1:Object]) : Object	Ja	Nein	Ja	Ja

FileReference

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die FileReference-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
creationDate:Date [read-only]	Ja	Nein	Nein	Nein
creator:String [schreibgeschützt]	Ja	Nein	Nein	Nein
modificationDate:Date [schreibgeschützt]	Ja	Nein	Nein	Nein
name:String [schreibgeschützt]	Ja	Nein	Nein	Nein
postData:String	Ja	Nein	Nein	Nein
size:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
type:String [schreibgeschützt]	Ja	Nein	Nein	Nein

FileReferenceList

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die FileReferenceList-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Ja	Nein	Nein	Nein
browse([typelist:Array]) : Boolean	Ja	Nein	Nein	Nein
fileList:Array	Ja	Nein	Nein	Nein
FileReferenceList()	Ja	Nein	Nein	Nein
onCancel = function(fileRefList:FileReferenceList) {}	Ja	Nein	Nein	Nein
onSelect = function(fileRefList:FileReferenceList) {}	Ja	Nein	Nein	Nein
removeListener(listener:Object) : Boolean	Ja	Nein	Nein	Nein

Function

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Function-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
apply(thisObject:Object, [argArray:Array]) : Void	Ja	Ja	Ja	Ja
call(thisObject:Object, [parameter1:Object]) : Object	Ja	Veraltet	Veraltet	Veraltet

GlowFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die GlowFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Ja	Nein	Nein	Nein
blurX:Number	Ja	Nein	Nein	Nein
blurY:Number	Ja	Nein	Nein	Nein
color:Number	Ja	Nein	Nein	Nein
inner:Boolean	Ja	Nein	Nein	Nein
knockout:Boolean	Ja	Nein	Nein	Nein
quality:Number	Ja	Nein	Nein	Nein
strength:Number	Ja	Nein	Nein	Nein

GradientBevelFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die GradientBevelFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alphas:Array	Ja	Nein	Nein	Nein
angle:Number	Ja	Nein	Nein	Nein
blurX:Number	Ja	Nein	Nein	Nein
blurY:Number	Ja	Nein	Nein	Nein
clone() : GradientBevelFilter	Ja	Nein	Nein	Nein
colors:Array	Ja	Nein	Nein	Nein
distance:Number	Ja	Nein	Nein	Nein
GradientBevelFilter([distance:Number], [angle:Number], [colors:Array], [alphas:Array], [ratios:Array], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [type:String], [knockout:Boolean])	Ja	Nein	Nein	Nein
knockout:Boolean	Ja	Nein	Nein	Nein
quality:Number	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
ratios:Array	Ja	Nein	Nein	Nein
strength:Number	Ja	Nein	Nein	Nein
type:String	Ja	Nein	Nein	Nein

GradientGlowFilter

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die GradientFlowFilter-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alphas:Array	Ja	Nein	Nein	Nein
angle:Number	Ja	Nein	Nein	Nein
blurX:Number	Ja	Nein	Nein	Nein
blurY:Number	Ja	Nein	Nein	Nein
colors:Array	Ja	Nein	Nein	Nein
distance:Number	Ja	Nein	Nein	Nein
GradientGlowFilter([distance:Number], [angle:Number], [colors:Array], [alphas:Array], [ratios:Array], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [type:String], [knockout:Boolean])	Ja	Nein	Nein	Nein
knockout:Boolean	Ja	Nein	Nein	Nein
quality:Number	Ja	Nein	Nein	Nein
ratios:Array	Ja	Nein	Nein	Nein
strength:Number	Ja	Nein	Nein	Nein
type:String	Ja	Nein	Nein	Nein

IME

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die IME-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Ja	Nein	Nein	Nein
ALPHANUMERIC_FULL:String	Ja	Nein	Nein	Nein
ALPHANUMERIC_HALF:String	Ja	Nein	Nein	Nein
CHINESE:String	Ja	Nein	Nein	Nein
doConversion() : Boolean	Ja	Nein	Nein	Nein
getConversionMode() : String	Ja	Nein	Nein	Nein
getEnabled() : Boolean	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
JAPANESE_HIRAGANA:String	Ja	Nein	Nein	Nein
JAPANESE_KATAKANA_FULL:String	Ja	Nein	Nein	Nein
JAPANESE_KATAKANA_HALF:String	Ja	Nein	Nein	Nein
KOREAN:String	Ja	Nein	Nein	Nein
onIMEComposition = function([readingString:String]) {}	Ja	Nein	Nein	Nein
removeListener(listener:Object) : Boolean	Ja	Nein	Nein	Nein
setCompositionString(composition:String) : Boolean	Ja	Nein	Nein	Nein
setConversionMode(mode:String) : Boolean	Ja	Nein	Nein	Nein
setEnabled(enabled:Boolean) : Boolean	Ja	Nein	Nein	Nein
UNKNOWN:String	Ja	Nein	Nein	Nein

Key

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Key-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_listeners:Array [schreibgeschützt]	Ja	Ja	Ja	Ja
addListener(listener:Object) : Void	Ja	Ja	Ja	Ja
BACKSPACE:Number	Ja	Ja	Ja	Ja
CAPSLOCK:Number	Ja	Ja	Ja	Ja
CONTROL:Number	Ja	Ja	Ja	Ja
DELETEKEY:Number	Ja	Ja	Ja	Ja
DOWN:Number	Ja	Ja	Ja	Ja
END:Number	Ja	Ja	Ja	Ja
ENTER:Number	Ja	Ja	Ja	Ja
ESCAPE:Number	Ja	Ja	Ja	Ja
getAscii() : Number	Ja	Ja	Ja	Ja
getCode() : Number	Ja	Ja	Ja	Ja
HOME:Number	Ja	Ja	Ja	Ja
INSERT:Number	Ja	Ja	Ja	Ja
isAccessible() : Boolean	Ja	Nein	Nein	Nein
isDown(code:Number) : Boolean	Ja	Ja	Ja	Ja
isToggled(code:Number) : Boolean	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
LEFT:Number	Ja	Ja	Ja	Ja
onKeyDown = function() {}	Ja	Ja	Ja	Ja
onKeyUp = function() {}	Ja	Ja	Ja	Ja
PGDN:Number	Ja	Ja	Ja	Ja
PGUP:Number	Ja	Ja	Ja	Ja
removeListener(listener:Object) : Boolean	Ja	Nein	Ja	Ja
RIGHT:Number	Ja	Ja	Ja	Ja
SHIFT:Number	Ja	Ja	Ja	Ja
SPACE:Number	Ja	Ja	Ja	Ja
TAB:Number	Ja	Ja	Ja	Ja
UP:Number	Ja	Ja	Ja	Ja

LoadVars

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die LoadVars-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addRequestHeader(header:Object, headerValue:String) : Void	Ja	Ja	Ja	Ja
contentType:String	Ja	Ja	Ja	Ja
decode(queryString:String) : Void	Ja	Ja	Ja	Ja
getBytesLoaded() : Number	Ja	Ja	Ja	Ja
getBytesTotal() : Number	Ja	Ja	Ja	Ja
load(url:String) : Boolean	Ja	Ja	Ja	Ja
loaded:Boolean	Ja	Ja	Ja	Ja
LoadVars()	Ja	Ja	Ja	Ja
onData = function(src:String) {}	Ja	Ja	Ja	Ja
onHTTPStatus = function(httpStatus:Number) {}	Ja	Nein	Nein	Ja
onLoad = function(success:Boolean) {}	Ja	Ja	Ja	Ja
send(url:String, target:String, [method:String]) : Boolean	Ja	Ja	Ja	Ja
sendAndLoad(url:String, target:Object, [method:String]) : Boolean	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja

LocalConnection

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die LocalConnection-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
allowDomain = function([sendingDomain:String]) {}	Ja	Nein	Nein	Ja
allowInsecureDomain = function([sendingDomain:String]) {}	Ja	Nein	Nein	Ja
close() : Void	Ja	Nein	Nein	Ja
connect(connectionName:String) : Boolean	Ja	Nein	Nein	Ja
domain() : String	Ja	Nein	Nein	Ja
LocalConnection()	Ja	Nein	Nein	Ja
onStatus = function(infoObject:Object) {}	Ja	Nein	Nein	Ja
send(connectionName:String, methodName:String, [args:Object]) : Boolean	Ja	Nein	Nein	Ja

Locale

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Locale-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addDelayedInstance(instance:Object, stringID:String) : Void	Ja	Nein	Nein	Nein
addXMLPath(langCode:String, path:String) : Void	Ja	Nein	Nein	Nein
autoReplace:Boolean	Ja	Nein	Nein	Nein
checkXMLStatus() : Boolean	Ja	Nein	Nein	Nein
getDefaultLang() : String	Ja	Nein	Nein	Nein
initialize() : Void	Ja	Nein	Nein	Nein
languageCodeArray:Array [read-only]	Ja	Nein	Nein	Nein
loadLanguageXML(xmlLanguageCode:String, customXmlCompleteCallback:Function) : Void	Ja	Nein	Nein	Nein
loadString(id:String) : String	Ja	Nein	Nein	Nein
loadStringEx(stringID:String, languageCode:String) : String	Ja	Nein	Nein	Nein
setDefaultLang(langCode:String) : Void	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
setLoadCallback(loadCallback:Function) : Void	Ja	Nein	Nein	Nein
setString(stringID:String, languageCode:String, stringValue:String) : Void	Ja	Nein	Nein	Nein
stringIDArray:Array [schreibgeschützt]	Ja	Nein	Nein	Nein

Math

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Math-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
abs(x:Number) : Number	Ja	Ja	Ja	Ja
acos(x:Number) : Number	Ja	Ja	Ja	Ja
asin(x:Number) : Number	Ja	Ja	Ja	Ja
atan(tangent:Number) : Number	Ja	Ja	Ja	Ja
atan2(y:Number, x:Number) : Number	Ja	Ja	Ja	Ja
ceil(x:Number) : Number	Ja	Ja	Ja	Ja
cos(x:Number) : Number	Ja	Ja	Ja	Ja
E:Number	Ja	Ja	Ja	Ja
exp(x:Number) : Number	Ja	Ja	Ja	Ja
floor(x:Number) : Number	Ja	Ja	Ja	Ja
LN10:Number	Ja	Ja	Ja	Ja
LN2:Number	Ja	Ja	Ja	Ja
log(x:Number) : Number	Ja	Ja	Ja	Ja
LOG10E:Number	Ja	Ja	Ja	Ja
LOG2E:Number	Ja	Ja	Ja	Ja
max(x:Number, y:Number) : Number	Ja	Ja	Ja	Ja
min(x:Number, y:Number) : Number	Ja	Ja	Ja	Ja
PI:Number	Ja	Ja	Ja	Ja
pow(x:Number, y:Number) : Number	Ja	Ja	Ja	Ja
random() : Number	Ja	Ja	Ja	Ja
round(x:Number) : Number	Ja	Ja	Ja	Ja
sin(x:Number) : Number	Ja	Ja	Ja	Ja
sqrt(x:Number) : Number	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
SQRT1_2:Number	Ja	Ja	Ja	Ja
SQRT2:Number	Ja	Ja	Ja	Ja
tan(x:Number) : Number	Ja	Ja	Ja	Ja

Matrix

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Matrix-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
a:Number	Ja	Nein	Nein	Ja
b:Number	Ja	Nein	Nein	Ja
c:Number	Ja	Nein	Nein	Ja
clone() : Matrix	Ja	Nein	Nein	Ja
concat(m:Matrix) : Void	Ja	Nein	Nein	Ja
createBox(scaleX:Number, scaleY:Number, [rotation:Number], [tx:Number], [ty:Number]) : Void	Ja	Nein	Nein	Ja
createGradientBox(width:Number, height:Number, [rotation:Number], [tx:Number], [ty:Number]) : Void	Ja	Nein	Nein	Nein
d:Number	Ja	Nein	Nein	Ja
deltaTransformPoint(pt:Point) : Point	Ja	Nein	Nein	Ja
identity() : Void	Ja	Nein	Nein	Ja
invert() : Void	Ja	Nein	Nein	Ja
Matrix([a:Number], [b:Number], [c:Number], [d:Number], [tx:Number], [ty:Number])	Ja	Nein	Nein	Ja
rotate(angle:Number) : Void	Ja	Nein	Nein	Ja
scale(sx:Number, sy:Number) : Void	Ja	Nein	Nein	Ja
toString() : String	Ja	Nein	Nein	Ja
transformPoint(pt:Point) : Point	Ja	Nein	Nein	Ja
translate(tx:Number, ty:Number) : Void	Ja	Nein	Nein	Ja
tx:Number	Ja	Nein	Nein	Ja
ty:Number	Ja	Nein	Nein	Ja

Microphone

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Microphone-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
activityLevel:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
gain:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
get([index:Number]) : Microphone	Ja	Nein	Nein	Nein
index:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
muted:Boolean [schreibgeschützt]	Ja	Nein	Nein	Nein
name:String [schreibgeschützt]	Ja	Nein	Nein	Nein
names:Array [schreibgeschützt]	Ja	Nein	Nein	Nein
onActivity = function(active:Boolean) {}	Ja	Nein	Nein	Nein
onStatus = function(infoObject:Object) {}	Ja	Nein	Nein	Nein
rate:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
setGain(gain:Number) : Void	Ja	Nein	Nein	Nein
setRate(rate:Number) : Void	Ja	Nein	Nein	Nein
setSilenceLevel(silenceLevel:Number, [timeOut:Number]) : Void	Ja	Nein	Nein	Nein
setUseEchoSuppression(useEchoSuppression:Boolean) : Void	Ja	Nein	Nein	Nein
silenceLevel:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
silenceTimeOut:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
useEchoSuppression:Boolean [schreibgeschützt]	Ja	Nein	Nein	Nein

Mouse

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Mouse-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Ja	Ja	Ja	Ja
hide() : Number	Ja	Nein	Nein	Nein
onMouseDown = function() {}	Ja	Ja	Ja	Ja
onMouseMove = function() {}	Ja	Ja	Ja	Ja
onMouseUp = function() {}	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
onMouseWheel = function([delta:Number], [scrollTarget:Object]) {}	Ja	Nein	Nein	Nein
removeListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
show() : Number	Ja	Nein	Nein	Nein

MovieClip

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die MovieClip-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Ja	Ja	Ja	Ja
_currentframe:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_droptarget:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_focusrect:Boolean	Ja	Ja	Ja	Ja
_framesloaded:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_height:Number	Ja	Ja	Ja	Ja
_highquality:Number	Veraltet	Veraltet	Veraltet	Veraltet
_lockroot:Boolean	Ja	Ja	Ja	Ja
_name:String	Ja	Ja	Ja	Ja
_parent:MovieClip	Ja	Ja	Ja	Ja
_quality:String	Ja	Ja	Ja	Ja
_rotation:Number	Ja	Ja	Ja	Ja
_soundbuftime:Number	Ja	Ja	Ja	Ja
_target:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_totalframes:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_url:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_visible:Boolean	Ja	Ja	Ja	Ja
_width:Number	Ja	Ja	Ja	Ja
_x:Number	Ja	Ja	Ja	Ja
_xmouse:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_xscale:Number	Ja	Ja	Ja	Ja
_y:Number	Ja	Ja	Ja	Ja
_ymouse:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_yscale:Number	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
attachAudio(id:Object) : Void	Ja	Nein	Nein	Nein
attachBitmap(bmp:BitmapData, depth:Number, [pixelSnapping:String], [smoothing:Boolean]) : Void	Ja	Nein	Nein	Ja
attachMovie(id:String, name:String, depth:Number, [initObject:Object]) : MovieClip	Ja	Ja	Ja	Ja
beginBitmapFill(bmp:BitmapData, [matrix:Matrix], [repeat:Boolean], [smoothing:Boolean]) : Void	Ja	Nein	Nein	Ja
beginFill(rgb:Number, [alpha:Number]) : Void	Ja	Ja	Ja	Ja
beginGradientFill(fillType:String, colors:Array, alphas:Array, ratios:Array, matrix:Object, [spreadMethod:String], [interpolationMethod:String], [focalPointRatio:Number]) : Void	Ja	Nein	Nein	Nein
blendMode:Object	Ja	Nein	Nein	Nein
cacheAsBitmap:Boolean	Ja	Nein	Nein	Nein
clear() : Void	Ja	Ja	Ja	Ja
createEmptyMovieClip(name:String, depth:Number) : MovieClip	Ja	Ja	Ja	Ja
createTextField(instanceName:String, depth:Number, x:Number, y:Number, width:Number, height:Number) : TextField	Ja	Ja	Ja	Ja
curveTo(controlX:Number, controlY:Number, anchorX:Number, anchorY:Number) : Void	Ja	Ja	Ja	Ja
duplicateMovieClip(name:String, depth:Number, [initObject:Object]) : MovieClip	Ja	Ja	Ja	Ja
enabled:Boolean	Ja	Ja	Ja	Ja
endFill() : Void	Ja	Ja	Ja	Ja
filters:Array	Ja	Nein	Nein	Nein
focusEnabled:Boolean	Ja	Ja	Ja	Ja
forceSmoothing:Boolean	Ja	Nein	Nein	Nein
getBounds(bounds:Object) : Object	Ja	Ja	Ja	Ja
getBytesLoaded() : Number	Ja	Ja	Ja	Ja
getBytesTotal() : Number	Ja	Ja	Ja	Ja
getDepth() : Number	Ja	Ja	Ja	Ja
getInstanceAtDepth(depth:Number) : MovieClip	Ja	Ja	Ja	Ja

Name der Funktion	**ActionScript 2**	**Flash Lite 2.1**	**Flash Lite 3.0**	**Flash Lite 3.1**
getNextHighestDepth() : Number	Ja	Ja	Ja	Ja
getRect(bounds:Object) : Object	Ja	Nein	Nein	Nein
getSWFVersion() : Number	Ja	Ja	Ja	Ja
getTextSnapshot() : TextSnapshot	Ja	Nein	Nein	Nein
getURL(url:String, [window:String], [method:String]) : Void	Ja	Ja	Ja	Ja
globalToLocal(pt:Object) : Void	Ja	Ja	Ja	Ja
gotoAndPlay(frame:Object) : Void	Ja	Ja	Ja	Ja
gotoAndStop(frame:Object) : Void	Ja	Ja	Ja	Ja
hitArea:Object	Ja	Ja	Ja	Ja
hitTest() : Boolean	Ja	Ja	Ja	Ja
lineGradientStyle(fillType:String, colors:Array, alphas:Array, ratios:Array, matrix:Object, [spreadMethod:String], [interpolationMethod:String], [focalPointRatio:Number]) : Void	Ja	Nein	Nein	Nein
lineStyle(thickness:Number, rgb:Number, alpha:Number, pixelHinting:Boolean, noScale:String, capsStyle:String, jointStyle:String, miterLimit:Number) : Void	Ja	Ja	Ja	Ja
lineTo(x:Number, y:Number) : Void	Ja	Ja	Ja	Ja
loadMovie(url:String, [method:String]) : Void	Ja	Ja	Ja	Ja
loadVariables(url:String, [method:String]) : Void	Ja	Ja	Ja	Ja
localToGlobal(pt:Object) : Void	Ja	Ja	Ja	Ja
menu:ContextMenu	Ja	Nein	Nein	Nein
moveTo(x:Number, y:Number) : Void	Ja	Ja	Ja	Ja
nextFrame() : Void	Ja	Ja	Ja	Ja
onData = function() {}	Ja	Ja	Ja	Ja
onDragOut = function() {}	Ja	Ja	Ja	Ja
onDragOver = function() {}	Ja	Ja	Ja	Ja
onEnterFrame = function() {}	Ja	Ja	Ja	Ja
onKeyDown = function() {}	Ja	Ja	Ja	Ja
onKeyUp = function() {}	Ja	Ja	Ja	Ja
onKillFocus = function(newFocus:Object) {}	Ja	Ja	Ja	Ja
onLoad = function() {}	Ja	Ja	Ja	Ja
onMouseDown = function() {}	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
onMouseMove = function() {}	Ja	Ja	Ja	Ja
onMouseUp = function() {}	Ja	Ja	Ja	Ja
onPress = function() {}	Ja	Ja	Ja	Ja
onRelease = function() {}	Ja	Ja	Ja	Ja
onReleaseOutside = function() {}	Ja	Ja	Ja	Ja
onRollOut = function() {}	Ja	Ja	Ja	Ja
onRollOver = function() {}	Ja	Ja	Ja	Ja
onSetFocus = function(oldFocus:Object) {}	Ja	Ja	Ja	Ja
onUnload = function() {}	Ja	Ja	Ja	Ja
opaqueBackground:Number	Ja	Nein	Nein	Nein
play() : Void	Ja	Ja	Ja	Ja
prevFrame() : Void	Ja	Ja	Ja	Ja
removeMovieClip() : Void	Ja	Ja	Ja	Ja
scale9Grid:Rectangle	Ja	Nein	Nein	Nein
scrollRect:Object	Ja	Nein	Nein	Nein
setMask(mc:Object) : Void	Ja	Ja	Ja	Ja
startDrag([lockCenter:Boolean], [left:Number], [top:Number], [right:Number], [bottom:Number]) : Void	Ja	Ja	Ja	Ja
stop() : Void	Ja	Ja	Ja	Ja
stopDrag() : Void	Ja	Ja	Ja	Ja
swapDepths(target:Object) : Void	Ja	Ja	Ja	Ja
tabChildren:Boolean	Ja	Ja	Ja	Ja
tabEnabled:Boolean	Ja	Ja	Ja	Ja
tabIndex:Number	Ja	Ja	Ja	Ja
trackAsMenu:Boolean	Ja	Ja	Ja	Ja
transform:Transform	Ja	Nein	Nein	Nein
unloadMovie() : Void	Ja	Ja	Ja	Ja
useHandCursor:Boolean	Ja	Nein	Nein	Nein

MovieClipLoader

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die MovieClipLoader-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
checkPolicyFile:Boolean	Ja	Nein	Nein	Nein
getProgress(target:Object) : Object	Ja	Ja	Ja	Ja
loadClip(url:String, target:Object) : Boolean	Ja	Ja	Ja	Ja
MovieClipLoader()	Ja	Ja	Ja	Ja
onLoadComplete = function([target_mc:MovieClip], [httpStatus:Number]) {}	Ja	Ja	Ja	Ja
onLoadError = function(target_mc:MovieClip, errorCode:String, [httpStatus:Number]) {}	Ja	Ja	Ja	Ja
onLoadInit = function([target_mc:MovieClip]) {}	Ja	Ja	Ja	Ja
onLoadProgress = function([target_mc:MovieClip], loadedBytes:Number, totalBytes:Number) {}	Ja	Ja	Ja	Ja
onLoadStart = function([target_mc:MovieClip]) {}	Ja	Ja	Ja	Ja
removeListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
unloadClip(target:Object) : Boolean	Ja	Ja	Ja	Ja

NetConnection

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die NetConnection-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
connect(targetURI:String) : Boolean	Ja	Nein	Ja	Ja
NetConnection()	Ja	Nein	Ja	Ja

NetStream

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die NetStream-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
bufferLength:Number [schreibgeschützt]	Ja	Nein	Ja	Ja
bufferTime:Number [schreibgeschützt]	Ja	Nein	Ja	Ja
bytesLoaded:Number [schreibgeschützt]	Ja	Nein	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
bytesTotal:Number [schreibgeschützt]	Ja	Nein	Ja	Ja
checkPolicyFile:Boolean	Ja	Nein	Nein	Nein
close() : Void	Ja	Nein	Ja	Ja
currentFps:Number [schreibgeschützt]	Ja	Nein	Ja	Ja
NetStream(connection:NetConnection)	Ja	Nein	Ja	Ja
onCuePoint = function(infoObject:Object) {}	Ja	Nein	Ja	Ja
onMetaData = function(infoObject:Object) {}	Ja	Nein	Ja	Ja
onStatus = function(infoObject:Object) {}	Ja	Nein	Ja	Ja
pause([flag:Boolean]) : Void	Ja	Nein	Ja	Ja
play(name:Object, start:Number, len:Number, reset:Object) : Void	Ja	Nein	Ja	Ja
seek(offset:Number) : Void	Ja	Nein	Ja	Ja
setBufferTime(bufferTime:Number) : Void	Ja	Nein	Ja	Ja
time:Number [schreibgeschützt]	Ja	Nein	Ja	Ja

Number

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Number-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
MAX_VALUE:Number	Ja	Ja	Ja	Ja
MIN_VALUE:Number	Ja	Ja	Ja	Ja
NaN:Number	Ja	Ja	Ja	Ja
NEGATIVE_INFINITY:Number	Ja	Ja	Ja	Ja
Number(num:Object)	Ja	Ja	Ja	Ja
POSITIVE_INFINITY:Number	Ja	Ja	Ja	Ja
toString(radix:Number) : String	Ja	Ja	Ja	Ja
valueOf() : Number	Ja	Ja	Ja	Ja

Object

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Object-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
__proto__:Object	Ja	Ja	Ja	Ja
__resolve:Object	Ja	Ja	Ja	Ja
addProperty(name:String, getter:Function, setter:Function) : Boolean	Ja	Ja	Ja	Ja
constructor:Object	Ja	Ja	Ja	Ja
hasOwnProperty(name:String) : Boolean	Ja	Ja	Ja	Ja
isPropertyEnumerable(name:String) : Boolean	Ja	Ja	Ja	Ja
isPrototypeOf(theClass:Object) : Boolean	Ja	Ja	Ja	Ja
Object()	Ja	Ja	Ja	Ja
prototype:Object	Ja	Ja	Ja	Ja
registerClass(name:String, theClass:Function) : Boolean	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja
unwatch(name:String) : Boolean	Ja	Ja	Ja	Ja
valueOf() : Object	Ja	Ja	Ja	Ja
watch(name:String,callback:Function, [userData:Object]) : Boolean	Ja	Ja	Ja	Ja

Point

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Point-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
add(v:Point) : Point	Ja	Nein	Nein	Ja
clone() : Point	Ja	Nein	Nein	Ja
distance(pt1:Point, pt2:Point) : Number	Ja	Nein	Nein	Ja
equals(toCompare:Object) : Boolean	Ja	Nein	Nein	Ja
interpolate(pt1:Point, pt2:Point, f:Number) : Point	Ja	Nein	Nein	Ja
length:Number	Ja	Nein	Nein	Ja
normalize(length:Number) : Void	Ja	Nein	Nein	Ja
offset(dx:Number, dy:Number) : Void	Ja	Nein	Nein	Ja
Point(x:Number, y:Number)	Ja	Nein	Nein	Ja
polar(len:Number, angle:Number) : Point	Ja	Nein	Nein	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
subtract(v:Point) : Point	Ja	Nein	Nein	Ja
toString() : String	Ja	Nein	Nein	Ja
x:Number	Ja	Nein	Nein	Ja
y:Number	Ja	Nein	Nein	Ja

PrintJob

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die PrintJob-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addPage(target:Object, [printArea:Object], [options:Object], [frameNum:Number]) : Boolean	Ja	Nein	Nein	Nein
orientation:String [schreibgeschützt]	Ja	Nein	Nein	Nein
pageHeight:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
pageWidth:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
paperHeight:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
paperWidth:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
PrintJob()	Ja	Nein	Nein	Nein
send() : Void	Ja	Nein	Nein	Nein
start() : Boolean	Ja	Nein	Nein	Nein

Rectangle

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Rectangle-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
bottom:Number	Ja	Nein	Nein	Ja
bottomRight:Point	Ja	Nein	Nein	Ja
clone() : Rectangle	Ja	Nein	Nein	Ja
contains(x:Number, y:Number) : Boolean	Ja	Nein	Nein	Ja
containsPoint(pt:Point) : Boolean	Ja	Nein	Nein	Ja
containsRectangle(rect:Rectangle) : Boolean	Ja	Nein	Nein	Ja
equals(toCompare:Object) : Boolean	Ja	Nein	Nein	Ja
height:Number	Ja	Nein	Nein	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
inflate(dx:Number, dy:Number) : Void	Ja	Nein	Nein	Ja
inflatePoint(pt:Point) : Void	Ja	Nein	Nein	Ja
intersection(toIntersect:Rectangle) : Rectangle	Ja	Nein	Nein	Ja
intersects(toIntersect:Rectangle) : Boolean	Ja	Nein	Nein	Ja
isEmpty() : Boolean	Ja	Nein	Nein	Ja
left:Number	Ja	Nein	Nein	Ja
offset(dx:Number, dy:Number) : Void	Ja	Nein	Nein	Ja
offsetPoint(pt:Point) : Void	Ja	Nein	Nein	Ja
Rectangle(x:Number, y:Number, width:Number, height:Number)	Ja	Nein	Nein	Ja
right:Number	Ja	Nein	Nein	Ja
setEmpty() : Void	Ja	Nein	Nein	Ja
size:Point	Ja	Nein	Nein	Ja
top:Number	Ja	Nein	Nein	Ja
topLeft:Point	Ja	Nein	Nein	Ja
toString() : String	Ja	Nein	Nein	Ja
union(toUnion:Rectangle) : Rectangle	Ja	Nein	Nein	Ja
width:Number	Ja	Nein	Nein	Ja
x:Number	Ja	Nein	Nein	Ja
y:Number	Ja	Nein	Nein	Ja

Sicherheit

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die security-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
allowDomain(domain1:String) : Void	Ja	Ja	Ja	Ja
allowInsecureDomain(domain:String) : Void	Ja	Ja	Ja	Ja
loadPolicyFile(url:String) : Void	Ja	Ja	Ja	Ja
sandboxType:String [schreibgeschützt]	Ja	Nein	Nein	Ja

Selection

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Selection-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Ja	Ja	Ja	Ja
getBeginIndex() : Number	Ja	Nein	Nein	Nein
getCaretIndex() : Number	Ja	Nein	Nein	Nein
getEndIndex() : Number	Ja	Nein	Nein	Nein
getFocus() : String	Ja	Ja	Ja	Ja
onSetFocus = function([oldfocus:Object], [newfocus:Object]) {}	Ja	Ja	Ja	Ja
removeListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
setFocus(newFocus:Object) : Boolean	Ja	Ja	Ja	Ja
setSelection(beginIndex:Number, endIndex:Number) : Void	Ja	Nein	Nein	Nein

SharedObject

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die SharedObject-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(objectName:String, notifyFunction:Function) : Void	Nein	Ja	Ja	Ja
clear() : Void	Ja	Ja	Ja	Ja
data:Object	Ja	Ja	Ja	Ja
flush([minDiskSpace:Number]) : Object	Ja	Teilweise	Teilweise	Teilweise
getLocal(name:String, [localPath:String], [secure:Boolean]) : SharedObject	Ja	Teilweise	Teilweise	Teilweise
getMaxSize() : Number	Nein	Ja	Ja	Ja
getSize() : Number	Ja	Ja	Ja	Ja
onStatus = function(infoObject:Object) {}	Ja	Ja	Ja	Ja
removeListener(objectName:String)	Nein	Ja	Ja	Ja

Sound

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Sound-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
attachSound(id:String) : Void	Ja	Ja	Ja	Ja
checkPolicyFile:Boolean	Ja	Nein	Nein	Nein
duration:Number [schreibgeschützt]	Ja	Teilweise	Teilweise	Teilweise

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getBytesLoaded() : Number	Ja	Ja	Ja	Ja
getBytesTotal() : Number	Ja	Ja	Ja	Ja
getPan() : Number	Ja	Teilweise	Teilweise	Teilweise
getTransform() : Object	Ja	Teilweise	Teilweise	Teilweise
getVolume() : Number	Ja	Ja	Ja	Ja
id3:Object [schreibgeschützt]	Ja	Ja	Ja	Ja
loadSound(url:String, isStreaming:Boolean) : Void	Ja	Teilweise	Teilweise	Teilweise
onID3 = function() {}	Ja	Ja	Ja	Ja
onLoad = function(success:Boolean) {}	Ja	Ja	Ja	Ja
onSoundComplete = function() {}	Ja	Ja	Ja	Ja
position:Number [schreibgeschützt]	Ja	Teilweise	Teilweise	Teilweise
setPan(value:Number) : Void	Ja	Teilweise	Teilweise	Teilweise
setTransform(transformObject:Object) : Void	Ja	Teilweise	Teilweise	Teilweise
setVolume(value:Number) : Void	Ja	Teilweise	Teilweise	Teilweise
Sound([target:Object])	Ja	Ja	Ja	Ja
start([secondOffset:Number], [loops:Number]) : Void	Ja	Ja	Ja	Ja
stop([linkageID:String]) : Void	Ja	Ja	Ja	Ja

Stage

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Stage-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Ja	Ja	Ja	Ja
align:String	Ja	Ja	Ja	Ja
displayState:String	Ja	Nein	Nein	Nein
height:Number	Ja	Ja	Ja	Ja
onFullScreen = function(bFull:Boolean) {}	Ja	Nein	Nein	Nein
onResize = function() {}	Ja	Ja	Ja	Ja
removeListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
scaleMode:String	Ja	Ja	Ja	Ja
showMenu:Boolean	Ja	Nein	Nein	Nein
width:Number	Ja	Ja	Ja	Ja

String

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die String-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
charAt(index:Number) : String	Ja	Ja	Ja	Ja
charCodeAt(index:Number) : Number	Ja	Ja	Ja	Ja
concat(value:Object) : String	Ja	Ja	Ja	Ja
fromCharCode() : String	Ja	Ja	Ja	Ja
indexOf(value:String, [startIndex:Number]) : Number	Ja	Ja	Ja	Ja
lastIndexOf(value:String, [startIndex:Number]) : Number	Ja	Ja	Ja	Ja
length:Number	Ja	Ja	Ja	Ja
slice(start:Number, end:Number) : String	Ja	Ja	Ja	Ja
split(delimiter:String, [limit:Number]) : Array	Ja	Ja	Ja	Ja
String(value:String)	Ja	Ja	Ja	Ja
substr(start:Number, length:Number) : String	Ja	Ja	Ja	Ja
substring(start:Number, end:Number) : String	Ja	Ja	Ja	Ja
toLowerCase() : String	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja
toUpperCase() : String	Ja	Ja	Ja	Ja
valueOf() : String	Ja	Ja	Ja	Ja

StyleSheet

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die StyleSheet-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
clear() : Void	Ja	Nein	Nein	Ja
getStyle(name:String) : Object	Ja	Nein	Nein	Ja
getStyleNames() : Array	Ja	Nein	Nein	Ja
load(url:String) : Boolean	Ja	Nein	Nein	Ja
onLoad = function(success:Boolean) {}	Ja	Nein	Nein	Ja
parseCSS(cssText:String) : Boolean	Ja	Nein	Nein	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
setStyle(name:String, style:Object) : Void	Ja	Nein	Nein	Ja
StyleSheet()	Ja	Nein	Nein	Ja
transform(style:Object) : TextFormat	Ja	Nein	Nein	Ja

System

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die System-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
exactSettings:Boolean	Ja	Nein	Nein	Nein
onStatus = function(infoObject:Object) {}	Ja	Ja	Ja	Ja
setClipboard(text:String) : Void	Ja	Nein	Nein	Nein
showSettings([tabID:Number]) : Void	Ja	Nein	Nein	Nein
useCodepage:Boolean	Ja	Ja	Ja	Ja

TextField

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die TextField-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Ja	Ja	Ja	Ja
_height:Number	Ja	Ja	Ja	Ja
_highquality:Number	Veraltet	Veraltet	Veraltet	Veraltet
_name:String	Ja	Ja	Ja	Ja
_parent:MovieClip	Ja	Ja	Ja	Ja
_quality:String	Ja	Ja	Ja	Ja
_rotation:Number	Ja	Ja	Ja	Ja
_soundbuftime:Number	Ja	Ja	Ja	Ja
_target:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_url:String [schreibgeschützt]	Ja	Ja	Ja	Ja
_visible:Boolean	Ja	Ja	Ja	Ja
_width:Number	Ja	Ja	Ja	Ja
_x:Number	Ja	Ja	Ja	Ja
_xmouse:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
_xscale:Number	Ja	Ja	Ja	Ja
_y:Number	Ja	Ja	Ja	Ja
_ymouse:Number [schreibgeschützt]	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_yscale:Number	Ja	Ja	Ja	Ja
addListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
antiAliasType:String	Ja	Nein	Nein	Nein
autoSize:Object	Ja	Ja	Ja	Ja
background:Boolean	Ja	Ja	Ja	Ja
backgroundColor:Number	Ja	Ja	Ja	Ja
border:Boolean	Ja	Ja	Ja	Ja
borderColor:Number	Ja	Ja	Ja	Ja
bottomScroll:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
condenseWhite:Boolean	Ja	Ja	Ja	Ja
embedFonts:Boolean	Ja	Ja	Ja	Ja
filters:Array	Ja	Nein	Nein	Nein
getDepth() : Number	Ja	Ja	Ja	Ja
getFontList() : Array	Ja	Nein	Nein	Nein
getNewTextFormat() : TextFormat	Ja	Ja	Ja	Ja
getTextFormat([beginIndex:Number], [endIndex:Number]) : TextFormat	Ja	Ja	Ja	Ja
gridFitType:String	Ja	Nein	Nein	Nein
hscroll:Number	Ja	Ja	Ja	Ja
html:Boolean	Ja	Ja	Ja	Ja
htmlText:String	Ja	Ja	Ja	Ja
length:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
maxChars:Number	Ja	Ja	Ja	Ja
maxhscroll:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
maxscroll:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
menu:ContextMenu	Ja	Nein	Nein	Nein
mouseWheelEnabled:Boolean	Ja	Nein	Nein	Nein
multiline:Boolean	Ja	Ja	Ja	Ja
onChanged = function(changedField:TextField) {}	Ja	Ja	Ja	Ja
onKillFocus = function(newFocus:Object) {}	Ja	Ja	Ja	Ja
onScroller = function(scrolledField:TextField) {}	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
onSetFocus = function(oldFocus:Object) {}	Ja	Ja	Ja	Ja
password:Boolean	Ja	Ja	Ja	Ja
removeListener(listener:Object) : Boolean	Ja	Ja	Ja	Ja
removeTextField() : Void	Ja	Ja	Ja	Ja
replaceSet(newText:String) : Void	Ja	Nein	Nein	Nein
replaceText(beginIndex:Number, endIndex:Number, newText:String) : Void	Ja	Ja	Ja	Ja
restrict:String	Ja	Nein	Nein	Nein
scroll:Number	Ja	Ja	Ja	Ja
selectable:Boolean	Ja	Ja	Ja	Ja
setNewTextFormat(tf:TextFormat) : Void	Ja	Ja	Ja	Ja
setTextFormat([beginIndex:Number], [endIndex:Number], textFormat:TextFormat) : Void	Ja	Ja	Ja	Ja
sharpness:Number	Ja	Nein	Nein	Nein
styleSheet:StyleSheet	Ja	Nein	Nein	Nein
tabEnabled:Boolean	Ja	Ja	Ja	Ja
tabIndex:Number	Ja	Ja	Ja	Ja
text:String	Ja	Ja	Ja	Ja
textColor:Number	Ja	Ja	Ja	Ja
textHeight:Number	Ja	Ja	Ja	Ja
textWidth:Number	Ja	Ja	Ja	Ja
thickness:Number	Ja	Nein	Nein	Nein
type:String	Ja	Ja	Ja	Ja
variable:String	Ja	Ja	Ja	Ja
wordWrap:Boolean	Ja	Ja	Ja	Ja

TextFormat

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die TextFormat-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
align:String	Ja	Ja	Ja	Ja
blockIndent:Number	Ja	Ja	Ja	Ja
bold:Boolean	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
bullet:Boolean	Ja	Teilweise	Teilweise	Teilweise
color:Number	Ja	Ja	Ja	Ja
font:String	Ja	Teilweise	Teilweise	Teilweise
getTextExtent(text:String, [width:Number]) : Object	Ja	Ja	Ja	Ja
indent:Number	Ja	Ja	Ja	Ja
italic:Boolean	Ja	Ja	Ja	Ja
kerning:Boolean	Ja	Nein	Nein	Nein
leading:Number	Ja	Ja	Ja	Ja
leftMargin:Number	Ja	Ja	Ja	Ja
letterSpacing:Number	Ja	Nein	Nein	Nein
rightMargin:Number	Ja	Ja	Ja	Ja
size:Number	Ja	Ja	Ja	Ja
tabStops:Array	Ja	Teilweise	Teilweise	Teilweise
target:String	Ja	Nein	Nein	Nein
TextFormat([font:String], [size:Number], [color:Number], [bold:Boolean], [italic:Boolean], [underline:Boolean], [url:String], [target:String], [align:String], [leftMargin:Number], [rightMargin:Number], [indent:Number], [leading:Number])	Ja	Ja	Ja	Ja
underline:Boolean	Ja	Ja	Ja	Ja
url:String	Ja	Nein	Nein	Nein

TextRenderer

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die TextRenderer-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
displayMode:String	Ja	Nein	Nein	Nein
maxLevel:Number	Ja	Nein	Nein	Nein
setAdvancedAntialiasingTable(fontName:String, fontStyle:String, colorType:String, advancedAntialiasingTable:Array) : Void	Ja	Nein	Nein	Nein

TextSnapshot

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die TextSnapshot-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
findText(startIndex:Number, textToFind:String, caseSensitive:Boolean) : Number	Ja	Nein	Nein	Nein
getCount() : Number	Ja	Nein	Nein	Nein
getSelected(start:Number, [end:Number]) : Boolean	Ja	Nein	Nein	Nein
getSelectedText([includeLineEndings: Boolean]) : String	Ja	Nein	Nein	Nein
getText(start:Number, end:Number, [includeLineEndings:Boolean]) : String	Ja	Nein	Nein	Nein
getTextRunInfo(beginIndex:Number, endIndex:Number) : Array	Ja	Nein	Nein	Nein
hitTestTextNearPos(x:Number, y:Number, [closeDist:Number]) : Number	Ja	Nein	Nein	Nein
setSelectColor(color:Number) : Void	Ja	Nein	Nein	Nein
setSelected(start:Number, end:Number, select:Boolean) : Void	Ja	Nein	Nein	Nein

Transform

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Transform-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
colorTransform:ColorTransform	Ja	Nein	Nein	Ja
concatenatedColorTransform:ColorTransform [schreibgeschützt]	Ja	Nein	Nein	Ja
concatenatedMatrix:Matrix [schreibgeschützt]	Ja	Nein	Nein	Ja
matrix:Matrix	Ja	Nein	Nein	Ja
pixelBounds:Rectangle	Ja	Nein	Nein	Ja
Transform(mc:MovieClip)	Ja	Nein	Nein	Ja

Video

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die Video-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Ja	Nein	Nein	Nein
_height:Number	Ja	Nein	Nein	Nein
_name:String	Ja	Nein	Nein	Nein
_parent:MovieClip	Ja	Nein	Nein	Nein
_rotation:Number	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_visible:Boolean	Ja	Nein	Nein	Nein
_width:Number	Ja	Nein	Nein	Nein
_x:Number	Ja	Nein	Nein	Nein
_xmouse:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
_xscale:Number	Ja	Nein	Nein	Nein
_y:Number	Ja	Nein	Nein	Nein
_ymouse:Number [schreibgeschützt]	Ja	Nein	Nein	Nein
_yscale:Number	Ja	Nein	Nein	Nein
attachVideo(source:Object) : Void	Ja	Nein	Ja	Ja
clear() : Void	Ja	Nein	Ja	Ja
close() : Void	Nein	Ja	Ja	Ja
deblocking:Number	Ja	Nein	Nein	Ja
height:Number [schreibgeschützt]	Ja	Nein	Ja	Ja
onStatus = function(infoObject:Object) {}	Nein	Ja	Ja	Ja
pause() : Void	Nein	Ja	Ja	Ja
play() : Boolean	Nein	Ja	Ja	Ja
resume() : Void	Nein	Ja	Ja	Ja
smoothing:Boolean	Ja	Nein	Nein	Ja
stop() : Void	Nein	Ja	Ja	Ja
width:Number [schreibgeschützt]	Ja	Nein	Ja	Ja

XML

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die XML-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addRequestHeader(header:Object, headerValue:String) : Void	Ja	Ja	Ja	Ja
contentType:String	Ja	Ja	Ja	Ja
createElement(name:String) : XMLNode	Ja	Ja	Ja	Ja
createTextNode(value:String) : XMLNode	Ja	Ja	Ja	Ja
docTypeDecl:String	Ja	Ja	Ja	Ja
getBytesLoaded() : Number	Ja	Ja	Ja	Ja
getBytesTotal() : Number	Ja	Ja	Ja	Ja
idMap:Object	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
ignoreWhite:Boolean	Ja	Ja	Ja	Ja
load(url:String) : Boolean	Ja	Ja	Ja	Ja
loaded:Boolean	Ja	Ja	Ja	Ja
onData = function(src:String) {}	Ja	Ja	Ja	Ja
onHTTPStatus = function(httpStatus:Number) {}	Ja	Nein	Nein	Ja
onLoad = function(success:Boolean) {}	Ja	Nein	Nein	Ja
parseXML(value:String) : Void	Ja	Ja	Ja	Ja
send(url:String, [target:String], [method:String]) : Boolean	Ja	Ja	Ja	Ja
sendAndLoad(url:String, resultXML:XML) : Void	Ja	Ja	Ja	Ja
status:Number	Ja	Ja	Ja	Ja
XML(text:String)	Ja	Ja	Ja	Ja
xmlDecl:String	Ja	Ja	Ja	Ja

XMLNode

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die XMLNode-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
appendChild(newChild:XMLNode) : Void	Ja	Ja	Ja	Ja
attributes:Object	Ja	Ja	Ja	Ja
childNodes:Array [schreibgeschützt]	Ja	Ja	Ja	Ja
cloneNode(deep:Boolean) : XMLNode	Ja	Ja	Ja	Ja
firstChild:XMLNode [schreibgeschützt]	Ja	Ja	Ja	Ja
getNamespaceForPrefix(prefix:String) : String	Ja	Nein	Nein	Nein
getPrefixForNamespace(nsURI:String) : String	Ja	Nein	Nein	Nein
hasChildNodes() : Boolean	Ja	Ja	Ja	Ja
insertBefore(newChild:XMLNode, insertPoint:XMLNode) : Void	Ja	Ja	Ja	Ja
lastChild:XMLNode [schreibgeschützt]	Ja	Ja	Ja	Ja
localName:String [schreibgeschützt]	Ja	Nein	Nein	Nein
namespaceURI:String [schreibgeschützt]	Ja	Nein	Nein	Nein

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
nextSibling:XMLNode [schreibgeschützt]	Ja	Ja	Ja	Ja
nodeName:String	Ja	Ja	Ja	Ja
nodeType:Number [schreibgeschützt]	Ja	Ja	Ja	Ja
nodeValue:String	Ja	Ja	Ja	Ja
parentNode:XMLNode [schreibgeschützt]	Ja	Ja	Ja	Ja
prefix:String [schreibgeschützt]	Ja	Nein	Nein	Nein
previousSibling:XMLNode [schreibgeschützt]	Ja	Ja	Ja	Ja
removeNode() : Void	Ja	Ja	Ja	Ja
toString() : String	Ja	Ja	Ja	Ja
XMLNode(type:Number, value:String)	Ja	Ja	Ja	Ja

XMLSocket

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die XMLSocket-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
close() : Void	Ja	Ja	Ja	Ja
connect(url:String, port:Number) : Boolean	Ja	Ja	Ja	Ja
onClose = function() {}	Ja	Ja	Ja	Ja
onConnect = function(success:Boolean) {}	Ja	Ja	Ja	Ja
onData = function(src:String) {}	Ja	Ja	Ja	Ja
onXML = function(src:XML) {}	Ja	Ja	Ja	Ja
send(data:Object) : Void	Ja	Ja	Ja	Ja
XMLSocket()	Ja	Ja	Ja	Ja

XMLUI

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die XMLUI-Klasse und ihre Mitglieder unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
accept() : Void	Ja	Nein	Nein	Nein
cancel() : Void	Ja	Nein	Nein	Nein
get(name:String) : String	Ja	Nein	Nein	Nein
set(name:String, value:String) : Void	Ja	Nein	Nein	Nein

Globale Funktionen und Eigenschaften

Die folgende Tabelle zeigt, welche Versionen von Flash Lite die verschiedenen globalen Funktionen und Eigenschaften unterstützen.

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
add	Nein	Ja	Ja	Ja
and	Nein	Ja	Ja	Ja
Array([numElements:Number], [elementN:Object])	Ja	Ja	Ja	Ja
asfunction(function:String, parameter:String)	Ja	Nein	Nein	Nein
Boolean(expression:Object)	Ja	Ja	Ja	Ja
break	Nein	Ja	Ja	Ja
call(frame:Object)	Veraltet	Veraltet	Veraltet	Veraltet
case	Nein	Ja	Ja	Ja
chr(number:Number)	Veraltet	Veraltet	Veraltet	Veraltet
clearInterval(intervalID:Number)	Ja	Ja	Ja	Ja
continue	Nein	Ja	Ja	Ja
default	Nein	Ja	Ja	Ja
do while	Nein	Ja	Ja	Ja
duplicateMovieClip(target:Object, newname:String, depth:Number)	Ja	Ja	Ja	Ja
else	Nein	Ja	Ja	Ja
else if	Nein	Ja	Ja	Ja
eq	Nein	Ja	Ja	Ja
escape(expression:String)	Ja	Ja	Ja	Ja
eval(expression:Object)	Ja	Ja	Ja	Ja
0	Nein	Ja	Ja	Ja
for	Nein	Ja	Ja	Ja
for … in	Nein	Ja	Ja	Ja
fscommand(command:String, parameters:String)	Ja	Ja	Ja	Ja
fscommand2(command:String, parameters:String)	Nein	Ja	Ja	Ja
Funktion	Nein	Ja	Ja	Ja
getProperty(my_mc:String, property)	Veraltet	Ja	Ja	Ja
getTimer()	Ja	Ja	Ja	Ja
getURL(url:String, [window:String], [method:String])	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getVersion()	Ja	Ja	Ja	Ja
gotoAndPlay([scene:String], frame:Object)	Ja	Ja	Ja	Ja
gotoAndStop([scene:String], frame:Object)	Ja	Ja	Ja	Ja
if	Nein	Ja	Ja	Ja
ifFrameLoaded([scene:String], frame:Object)	Veraltet	Veraltet	Veraltet	Veraltet
int(value:Number)	Veraltet	Veraltet	Veraltet	Veraltet
isFinite(expression:Object)	Ja	Ja	Ja	Ja
isNaN(expression:Object)	Ja	Ja	Ja	Ja
length(expression:String, variable:Object)	Veraltet	Veraltet	Veraltet	Veraltet
loadMovie(url:String, target:Object, [method:String])	Ja	Ja	Ja	Ja
loadMovieNum(url:String, level:Number, [method:String])	Ja	Ja	Ja	Ja
loadVariables(url:String, target:Object, [method:String])	Ja	Ja	Ja	Ja
loadVariablesNum(url:String, level:Number, [method:String])	Ja	Ja	Ja	Ja
mbchr(number:Number)	Veraltet	Veraltet	Veraltet	Veraltet
mblength(string:String)	Veraltet	Veraltet	Veraltet	Veraltet
mbord(character:String)	Veraltet	Veraltet	Veraltet	Veraltet
mbsubstring(value:, index:Number, count:Number)String	Veraltet	Veraltet	Veraltet	Veraltet
MMExecute(command:String)	Ja	Nein	Nein	Nein
ne	Nein	Ja	Ja	Ja
new	Nein	Ja	Ja	Ja
newline	Nein	Ja	Ja	Ja
nextFrame()	Ja	Ja	Ja	Ja
nextScene()	Ja	Ja	Ja	Ja
und nicht etwa	Nein	Ja	Ja	Ja
Number(expression:Object)	Ja	Ja	Ja	Ja
Object([value:Object])	Ja	Ja	Ja	Ja
on(mouseEvent:Object)	Ja	Ja	Ja	Ja
onClipEvent(movieEvent:Object)	Ja	Ja	Ja	Ja
ord(character:String)	Veraltet	Veraltet	Veraltet	Veraltet

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
parseFloat(string:String)	Ja	Ja	Ja	Ja
parseInt(expression:String, [radix:Number])	Ja	Ja	Ja	Ja
play()	Ja	Ja	Ja	Ja
prevFrame()	Ja	Ja	Ja	Ja
prevScene()	Ja	Ja	Ja	Ja
print(target:Object, boundingBox:String)	Ja	Nein	Nein	Nein
printAsBitmap(target:Object, boundingBox:String)	Ja	Nein	Nein	Nein
printAsBitmapNum(level:Number, boundingBox:String)	Ja	Nein	Nein	Nein
printNum(level:Number, boundingBox:String)	Ja	Nein	Nein	Nein
random(value:Number)	Veraltet	Veraltet	Veraltet	Veraltet
removeMovieClip(target:Object)	Ja	Ja	Ja	Ja
return	Nein	Ja	Ja	Ja
set	Nein	Ja	Ja	Ja
setInterval(functionReference:Function, interval:Number, [param:Object], objectReference:Object, methodName:String)	Ja	Ja	Ja	Ja
setProperty(target:Object, property:Object, expression:Object)	Ja	Ja	Ja	Ja
showRedrawRegions(enable:Boolean, [color:Number])	Ja	Nein	Nein	Nein
startDrag(target:Object, [lock:Boolean], [left,top,right,bottom:Number])	Ja	Ja	Ja	Ja
stop()	Ja	Ja	Ja	Ja
stopAllSounds()	Ja	Ja	Ja	Ja
stopDrag()	Ja	Ja	Ja	Ja
String(expression:Object)	Ja	Ja	Ja	Ja
substring(:String, index:Number, count:Number)String	Veraltet	Veraltet	Veraltet	Veraltet
switch	Nein	Ja	Ja	Ja
targetPath(targetObject:Object)	Ja	Ja	Ja	Ja
tellTarget(target:String, statement(s))	Veraltet	Veraltet	Veraltet	Veraltet
toggleHighQuality()	Veraltet	Veraltet	Veraltet	Veraltet
trace(expression:Object)	Ja	Ja	Ja	Ja

Name der Funktion	ActionScript 2	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
unescape(string:String)	Ja	Ja	Ja	Ja
unloadMovie(target:Object)	Ja	Ja	Ja	Ja
unloadMovieNum(level:Number)	Ja	Ja	Ja	Ja
updateAfterEvent()	Ja	Nein	Nein	Nein

Nicht und teilweise unterstützte Klassen: Details

In diesem Kapitel werden ActionScript 2.0-Klassen beschrieben, die von Macromedia Flash Lite 2.0 von Adobe, Adobe Flash Lite 2.1 und Adobe Flash Lite 3.x. nicht oder nur teilweise unterstützt werden. Darüber hinaus werden die speziellen Erweiterungen in ActionScript für Flash Lite 2.x und 3.x beschrieben.

Button-Klasse

Alle Schaltflächensymbole in einer SWF-Datei sind Instanzen des Button-Objekts. Die Button-Klasse enthält Methoden, Eigenschaften und Ereignisprozeduren zum Verwenden von Schaltflächen.

Weitere Informationen zur Button-Klasse finden Sie in folgenden Quellen:

- Kapitel 10, „Ereignisse verarbeiten“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften aufgeführt, die in ActionScript für Flash Lite 2.x und 3.x von der Button-Klasse nicht oder nur teilweise unterstützt werden.

Eigenschaft	Beschreibung	Unterstützung
`menu`	Ein Objekt, mit dem ein ContextMenu-Objekt mit einer Schaltfläche verknüpft wird.	Nicht unterstützt
`trackAsMenu`	Ein boolescher Wert, der angibt, ob andere Schaltflächen die durch das Loslassen der Maustaste erzeugten Ereignisse empfangen können. Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`useHandCursor`	Ein boolescher Wert, der angibt, ob der Mauszeiger angezeigt wird, wenn die Maus über eine Schaltfläche bewegt wird.	Nicht unterstützt
`_xmouse`	Schreibgeschützt. Die *x*-Koordinate des Mauszeigers relativ zu einer Schaltflächeninstanz. Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`_ymouse`	Schreibgeschützt. Die *y*-Koordinate des Mauszeigers relativ zu einer Schaltflächeninstanz. Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt

Übersicht über Ereignisse

In der folgenden Tabelle sind die Ereignisprozeduren aufgeführt, die in ActionScript für Flash Lite 2.x und 3.x von der Button-Klasse nur teilweise unterstützt werden.

Ereignis	Beschreibung
`onDragOut`	Wird aufgerufen, wenn die Maustaste über der Schaltfläche geklickt und der Mauszeiger dann aus der Schaltfläche bewegt wird. **Einschränkungen:**Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.
`onDragOver`	Wird aufgerufen, wenn der Mauszeiger über die Schaltfläche bewegt wird. **Einschränkungen:**Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.
`onReleaseOutside`	Wird aufgerufen, wenn die Maustaste losgelassen wird, während sich der Mauszeiger außerhalb der Schaltfläche befindet, nachdem die Maustaste gedrückt wurde, während sich der Mauszeiger innerhalb der Schaltfläche befand. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.

Date-Klasse

Die Date-Klasse dient der Darstellung von Datums- und Uhrzeitangaben in ActionScript. Sie unterstützt Vorgänge, mit denen Datum und Uhrzeit bearbeitet werden können. Mithilfe der Date-Klasse können auch das aktuelle Datum und die aktuelle Uhrzeit vom Betriebssystem abgerufen werden.

Weitere Informationen zur Date-Klasse finden Sie in folgenden Quellen:

- Kapitel 4, „Daten und Datentypen“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden (Funktionen) aufgeführt, die in ActionScript für Flash Lite 2.x und 3.x der Date-Klasse hinzugefügt wurden.

Methode	Beschreibung
`getLocaleLongDate()`	Diese Funktion legt einen Parameter für einen String fest, der das aktuelle Datum im langen Format angibt. Das Format entspricht dem derzeit definierten Gebietsschema. Der Parameter wird anhand des Namens übergeben. Der zurückgegebene Wert ist ein String mit mehreren Zeichen und variabler Länge. Die tatsächliche Formatierung hängt vom Gerät und vom Gebietsschema ab.
`getLocaleShortDate()`	Diese Funktion legt einen Parameter für einen String fest, der das aktuelle Datum im kurzen Format angibt. Das Format entspricht dem derzeit definierten Gebietsschema. Der Parameter wird anhand des Namens übergeben. Der zurückgegebene Wert ist ein String mit mehreren Zeichen und variabler Länge. Die tatsächliche Formatierung hängt vom Gerät und vom Gebietsschema ab.
`getLocaleTime()`	Diese Funktion legt einen Parameter für einen String fest, der die aktuelle Uhrzeit angibt. Das Format entspricht dem derzeit definierten Gebietsschema. Der Parameter wird anhand des Namens übergeben. Der zurückgegebene Wert ist ein String mit mehreren Zeichen und variabler Länge. Die tatsächliche Formatierung hängt vom Gerät und vom Gebietsschema ab.

Key-Klasse

Die Key-Klasse enthält Methoden und Eigenschaften zum Abrufen von Informationen über die Tastatur und über gedrückte Tasten auf der Tastatur.

Weitere Informationen zur Key-Klasse finden Sie in folgenden Quellen:

- Kapitel 14, „Interaktionen mit ActionScript erstellen“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Zusätzliche Eigenschaften in Flash Lite

In der folgenden Tabelle ist die Eigenschaft aufgeführt, die in ActionScript für Flash Lite 2.x und später zur Key-Klasse hinzugefügt wurde.

Eigenschaft	Beschreibung	Unterstützung
`isToggled`	Gibt `true` zurück, wenn die Taste Num oder die Feststelltaste aktiviert ist.	Nicht unterstützt

Zusätzliche Methoden in Flash Lite

In der folgenden Tabelle ist die zusätzliche Methode aufgeführt, die in ActionScript für Flash Lite 2.x und später der Key-Klasse hinzugefügt wurde.

Methode	Beschreibung
`getCode()`	Gibt den virtuellen Tastencode der zuletzt gedrückten Taste zurück.

Die Flash Lite 2.x- und 3.0.x-Implementierung der `getCode()`-Methode gibt je nach dem von der Plattform übergebenen Wert einen String oder eine Zahl zurück. Die zulässigen Tastencodes sind nur die von dieser Klasse akzeptierten Standardtastencodes sowie die "Sondercodes", die als Eigenschaften der ExtendedKey-Klasse aufgeführt sind. Diese Einschränkung wird durch den Player vorgegeben. Eine Beschreibung der gültigen Tastencodewerte finden Sie im Abschnitt zur Key-Klasse im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*. In diesem Referenzhandbuch sind Tabellen mit Tastenzuordnungen von Codes für Buchstaben, Ziffern, den numerischen Ziffernblock, Funktionstasten, spezielle Tastaturkonstanten und andere Tasten aufgeführt.

Mouse-Klasse

Mit der Mouse-Klasse können Sie die Maus in einer SWF-Datei steuern. So können Sie beispielsweise den Mauszeiger ein- oder ausblenden.

Weitere Informationen zur Mouse-Klasse finden Sie in folgenden Quellen:

- Kapitel 14, „Interaktionen mit ActionScript erstellen“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der Mouse-Klasse nicht oder nur teilweise unterstützt werden.

Methode	**Beschreibung**	**Unterstützung**
`addListener()`	Registriert ein Objekt zum Empfang von Benachrichtigungen der Listener `onMouseDown`, `onMouseMove`, `onMouseUp` und `onMouseWheel`. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`hide()`	Blendet den Mauszeiger in einer SWF-Datei aus.	Nicht unterstützt
`removeListener()`	Entfernt ein Objekt, das zuvor mit `addListener()` registriert wurde. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`show()`	Blendet den Mauszeiger in einer SWF-Datei ein.	Nicht unterstützt

Übersicht über Ereignisse

In der folgenden Tabelle sind die Ereignisse aufgeführt, die in ActionScript für Flash Lite 2.x und später von der Mouse-Klasse nicht oder nur teilweise unterstützt werden.

Ereignis	**Beschreibung**	**Unterstützung**
`onMouseDown`	Benachrichtigung beim Drücken einer Maustaste. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`onMouseMove`	Benachrichtigung bei Mausbewegung. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` auf `true` gesetzt ist.	Teilweise unterstützt
`onMouseUp`	Benachrichtigung beim Loslassen einer Maustaste. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`onMouseWheel`	Benachrichtigung erfolgt, wenn der Benutzer das Mausrad dreht.	Nicht unterstützt

MovieClip-Klasse

Mit der MovieClip-Klasse können Sie Listener-Rückruffunktionen verwenden, die beim Laden (Herunterladen) von SWF- oder JPEG-Dateien in Movieclips Statusinformationen zurückgeben. Setzen Sie zur Verwendung von MovieClip-Funktionen beim Laden von SWF-Dateien anstelle von `loadMovie()` oder `MovieClip.loadMovie()` den Aufruf `MovieClipLoader.loadClip()` ein.

Weitere Informationen zur MovieClip-Klasse finden Sie in folgenden Quellen:

- Kapitel 11, „Mit Movieclips arbeiten“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der MovieClip-Klasse nicht oder nur teilweise unterstützt werden.

Methode	Beschreibung	Unterstützung
`attachAudio()`	Zeichnet lokale Audiodaten über die Mikrofonhardware des Geräts auf und gibt sie wieder.	Nicht unterstützt
`getTextSnapshot()`	Gibt ein TextSnapshot-Objekt zurück, das den Text in den statischen Textfeldern des angegebenen Movieclips enthält.	Nicht unterstützt
`startDrag()`	Definiert einen Movieclip als ziehbar und startet den Ziehvorgang des Movieclips. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`stopDrag()`	Beendet eine `MovieClip.startDrag()`-Methode. Ein Movieclip, der mit `startDrag()` ziehbar gemacht wurde, bleibt so lange ziehbar, bis eine `stopDrag()`-Methode aufgerufen oder ein anderer Movieclip ziehbar gemacht wird. Nur jeweils ein Movieclip kann ziehbar sein. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften aufgeführt, die in ActionScript für Flash Lite 2.x und später von der MovieClip-Klasse nicht oder nur teilweise unterstützt werden.

Eigenschaft	Beschreibung	Unterstützung
`_droptarget`	Gibt den absoluten Pfad der Movieclip-Instanz, auf der dieser Movieclip abgelegt wurde, in Schrägstrichsyntax zurück. Die `_droptarget`-Eigenschaft gibt immer einen Pfad zurück, der mit einem Schrägstrich (/) beginnt. Um die `_droptarget`-Eigenschaft einer Instanz mit einer Referenz zu vergleichen, wandeln Sie den Rückgabewert, der die Schrägstrichsyntax aufweist, mit der `eval()`-Funktion in eine Referenz mit Punktsyntax um. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`menu`	Verknüpft das angegebene ContextMenu-Objekt mit dem Movieclip.	Nicht unterstützt
`_quality`	Stellt die für eine SWF-Datei verwendete Wiedergabequalität ein bzw. ruft diese ab. Da für Geräteschriftarten immer ein Aliasing erfolgt, hat die Eigenschaft `_quality` auf diese keine Auswirkungen.	Teilweise unterstützt

Eigenschaft	**Beschreibung**	**Unterstützung**
`trackAsMenu`	Ein boolescher Wert, der angibt, ob andere Schaltflächen oder Movieclips die durch das Loslassen der Maustaste erzeugten Ereignisse empfangen können. Die `trackAsMenu`-Eigenschaft dient zum Erstellen von Menüs. Die `trackAsMenu`-Eigenschaft kann für jedes beliebige Button- oder MovieClip-Objekt festgelegt werden. Wenn die `trackAsMenu`-Eigenschaft nicht vorhanden ist, lautet das Standardverhalten `false`. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`useHandCursor`	Ein boolescher Wert, der angibt, ob das Handsymbol angezeigt wird, wenn die Maus über einen Movieclip bewegt wird.	Nicht unterstützt
`_xmouse`	Gibt die x-Koordinate der Mausposition zurück. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt
`_ymouse`	Gibt die y-Koordinate der Mausposition zurück. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.	Teilweise unterstützt

Übersicht über Ereignisse

In der folgenden Tabelle sind die Ereignisprozeduren aufgeführt, die in ActionScript für Flash Lite 2.x und später von der MovieClip-Klasse nur teilweise unterstützt werden.

Ereignisprozedur	**Beschreibung**
`onDragOut`	Wird aufgerufen, wenn die Maustaste gedrückt und der Mauszeiger dann aus dem Objekt bewegt wird. Sie müssen eine Funktion definieren, die beim Aufruf der Ereignisprozedur ausgeführt wird. Sie können die Funktion auf der Zeitleiste oder in einer Klassendatei definieren, die die MovieClip-Klasse erweitert oder mit einem Symbol in der Bibliothek verknüpft ist. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.
`onDragOver`	Wird aufgerufen, wenn die Maustaste außerhalb des Movieclips gedrückt und der Mauszeiger dann über den Movieclip bewegt wird. Sie müssen eine Funktion definieren, die beim Aufruf der Ereignisprozedur ausgeführt wird. Sie können die Funktion auf der Zeitleiste oder in einer Klassendatei definieren, die die MovieClip-Klasse erweitert oder mit einem Symbol in der Bibliothek verknüpft ist. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.
`onMouseDown`	Wird aufgerufen, wenn die linke Maustaste gedrückt wird. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.

Ereignisprozedur	Beschreibung
`onMouseMove`	Wird bei jeder Mausbewegung aufgerufen. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` auf `true` gesetzt ist.
`onMouseUp`	Wird aufgerufen, wenn die linke Maustaste gedrückt wird. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.
`onReleaseOutside`	Wird aufgerufen, wenn die Maustaste gedrückt wird, während sich der Mauszeiger über einem Movieclip befindet und anschließend außerhalb des Movieclip-Bereichs wieder losgelassen wird. **Einschränkungen:** Wird unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` auf `true` gesetzt ist.

NetConnection-Klasse

Mithilfe der NetConnection-Klasse können Sie ein Objekt erstellen, das Sie mit einem NetStream-Objekt zum Aufrufen von Befehlen auf einem Remote-Anwendungsserver oder zum Wiedergeben von Streaming-Flash Video-Dateien (FLV) entweder lokal oder auf einem Server verwenden können. Unterstützung für diese Klasse wurde in Flash Lite 3.0 hinzugefügt.

Weitere Informationen zur NetConnection-Klasse finden Sie in folgenden Quellen:

- „Mit Bildern, Audio und Video arbeiten“ im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

NetStream-Klasse

Mithilfe der NetStream-Klasse können Sie einen Stream erstellen, der mit einem NetConnection-Objekt zum Wiedergeben von FLV-Dateien von einem lokalen Dateisystem, einer HTTP-Adresse oder einem Flash Media Server aus verwendet werden kann. Unterstützung für diese Klasse wurde in Flash Lite 3.0 hinzugefügt.

Weitere Informationen zur NetStream-Klasse finden Sie in folgenden Quellen:

- „Mit Bildern, Audio und Video arbeiten“ im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Eigenschaften

In der folgenden Tabelle ist die Eigenschaft aufgeführt, die in ActionScript für Flash Lite 3.0 und später von der NetSream-Klasse nicht unterstützt wird.

Eigenschaft	Beschreibung
`checkPolicyFile`	Gibt an, ob der Player vor dem Laden der eigentlichen FLV-Datei versuchen soll, eine domänenübergreifende Richtliniendatei von dem Server der geladenen FLV-Datei herunterzuladen.

Selection-Klasse

Mit der Selection-Klasse können Sie festlegen und steuern, in welchem Textfeld sich die Einfügemarke befindet (d. h., welches Feld den Fokus erhalten soll).

In Flash Lite 2.1 wird die Inline-Texteingabe durch die Selection-Klasse teilweise unterstützt. In Flash Lite ist das Selection-Objekt nur dann gültig, wenn ein Gerät die Inline-Texteingabe unterstützt. Unterstützt ein Gerät die Inline-Texteingabe nicht und verlässt es sich zur Eingabe von Text stattdessen auf einen FEP (Frontend-Prozessor), werden alle Aufrufe des Selection-Objekts ignoriert.

Weitere Informationen zur Selection-Klasse finden Sie in Kapitel 2, "ActionScript-Klassen", im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der Selection-Klasse nicht unterstützt werden.

Methode	Beschreibung
`getBeginIndex()`	Gibt den Index am Anfang des Auswahlbereichs zurück, der momentan den Fokus besitzt.
`getCaretIndex()`	Gibt den Index der blinkenden Einfügemarke zurück.
`getEndIndex`	Gibt den Index am Ende des Auswahlbereichs zurück, der momentan den Fokus besitzt.

SharedObject-Klasse

Mit einem gemeinsamen Objekt in Flash Lite, das mit der SharedObject-Klasse von ActionScript implementiert wird, kann Flash Lite-Inhalt beim Schließen einer Anwendung Daten auf dem Gerät speichern und diese Daten beim erneuten Ausführen der Anwendung vom Gerät laden.

Weitere Informationen zur SharedObject-Klasse finden Sie in folgenden Quellen:

- Kapitel 14, „Interaktionen mit ActionScript erstellen“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der SharedObject-Klasse nur teilweise unterstützt werden.

Methode	Beschreibung
`flush()`	Schreibt ein lokal permanentes gemeinsames Objekt in eine lokale Datei. **Einschränkungen:** Der Schreibvorgang ist asynchron und das Ergebnis nicht sofort verfügbar.
`getLocal()`	Gibt einen Verweis auf ein lokal permanentes gemeinsames Objekt zurück, das nur für den aktuellen Client verfügbar ist. **Einschränkungen:** In Flash Lite kann ein gemeinsames Objekt nicht von zwei verschiedenen SWF-Dateien verwendet werden.

Zusätzliche Methoden in Flash Lite

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später als Erweiterungen zur SharedObject-Klasse hinzugefügt wurden.

Methode	Beschreibung
`GetMaxSize()`	Flash Lite gibt den für eine SWF-Datei maximal zugeordneten Speicherplatz zur permanenten Datenspeicherung zurück.

System-Klasse

Die System-Klasse enthält Eigenschaften für bestimmte auf dem Computer des Benutzers durchgeführte Operationen, beispielsweise mit gemeinsamen Objekten, lokalen Einstellungen für Kameras und Mikrofone sowie der gemeinsamen Verwendung der Zwischenablage. Weitere Eigenschaften und Methoden befinden sich in spezifischen Klassen im System-Paket: die capabilities-Klasse, die security-Klasse und die IME-Klasse.

Weitere Informationen zur System-Klasse finden Sie in folgenden Quellen:

- *Flash Lite 2.x -und 3.x-Anwendungen entwickeln*
- *Flash Lite 2.x und 3 ActionScript-Referenzhandbuch*
- Kapitel 7, „Klassen“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der System-Klasse nicht unterstützt werden.

Methode	Beschreibung	Unterstützung
`setClipboard()`	Ersetzt den Inhalt der Zwischenablage durch einen festgelegten Textstring.	Nicht unterstützt
`showSettings()`	Zeigt das angegebene Bedienfeld für die Flash Player-Einstellungen an.	Nicht unterstützt

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften aufgeführt, die in ActionScript für Flash Lite 2.x und später von der System-Klasse nicht unterstützt werden.

Eigenschaft	Beschreibung	Unterstützung
`exactSettings`	Gibt an, ob für den Zugriff auf lokale Einstellungen die Regeln für die Übereinstimmung von Superdomänen oder die Regeln für die exakte Übereinstimmung von Domänen verwendet werden sollen.	Nicht unterstützt

Übersicht über Ereignisse

In der folgenden Tabelle sind die Ereignisprozeduren aufgeführt, die in ActionScript für Flash Lite 2.x und später von der System-Klasse nur teilweise unterstützt werden.

Ereignisprozedur	Beschreibung	Unterstützung
`onStatus`	Eine übergeordnete Ereignisprozedur für bestimmte Objekte. Die SharedObject-Eigenschaft wird unterstützt. Die LocalConnection- und die NetStream-Eigenschaft werden nicht unterstützt.	Teilweise unterstützt

capabilities-Klasse (System.capabilities)

Die System.capabilities-Klasse beschreibt den Funktionsumfang des Systems und des Players, auf denen eine SWF-Datei wiedergegeben wird. Auf diese Weise können Sie Inhalte für unterschiedliche Formate definieren. Das Display eines Mobiltelefons (schwarzweiß, 100 Quadratpixel groß) unterscheidet sich beispielsweise vom 1000 Quadratpixel großen Farbbildschirm eines PCs. Um benutzergerechte Inhalte möglichst vielen Benutzern bereitzustellen, können Sie mit dem System.capabilities-Objekt ermitteln, welches System ein bestimmter Benutzer verwendet. Sie können dann den Server auffordern, je nach Gerätekonfiguration unterschiedliche SWF-Dateien zu übertragen, oder die SWF-Datei anweisen, die Darstellung entsprechend zu ändern.

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften aufgeführt, die in ActionScript für Flash Lite 2.x und später von der System.capabilities-Klasse nicht unterstützt werden.

Eigenschaft	Beschreibung
`hasIME`	Gibt an, ob ein Eingabemethodeneditor (Input Method Editor - IME) auf dem System installiert ist.
`manufacturer`	Ein String, der den Hersteller von Flash Player im Format „Abobe OSName" angibt (OSName kann für „Windows", „Macintosh", „Linux" oder „Other OS Name" stehen).
`pixelAspectRatio`	Gibt das Pixelseitenverhältnis des Bildschirms an.
`playerType`	Gibt den Typ des Players an: eigenständig, extern, Plug-In oder ActiveX.
`screenColor`	Gibt die Darstellung des Bildschirms an: Farbe, Graustufen oder Schwarzweiß.
`screenDPI`	Gibt die Bildschirmauflösung in Pixel pro Zoll (dpi) an.
`serverString`	Ein URL-kodierter String, der die Werte für jede `System.capabilities`-Eigenschaft angibt.

Zusätzliche Eigenschaften in Flash Lite

In der folgenden Tabelle sind Eigenschaften aufgeführt, die in ActionScript für Flash Lite 2.x und später als Erweiterungen der System.capabilities-Klasse verwendet werden.

Eigenschaft	Beschreibung
`hasCompoundSound`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player zusammengesetzte Sounddaten verarbeiten kann. Andernfalls ist der Wert auf `false` gesetzt.
`hasEmail`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player E-Mail-Nachrichten mit dem ActionScript-Befehl `GetURL` senden kann. Andernfalls ist der Wert auf `false` gesetzt.
`hasMMS`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player MMS-Nachrichten (Multimedia Message Service) mit dem ActionScript-Befehl `GetURL` senden kann. Andernfalls ist der Wert auf `false` gesetzt.
`hasSMS`	Ein schreibgeschützter Wert, dessen Variable ***`_capSMS`*** angibt, ob Flash Lite SMS-Nachrichten (Short Message Service) mit dem ActionScript-Befehl `GetURL` senden kann. Wenn Flash Lite SMS-Nachrichten senden kann, ist diese Variable definiert und hat den Wert 1. Andernfalls ist die Variable nicht definiert.
`hasMFI`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player Sounddaten im Melodieformat für i-mode **(**MFi) wiedergeben kann. Andernfalls ist der Wert auf `false` gesetzt.

Eigenschaft	Beschreibung
`hasMIDI`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player Sounddaten im MIDI-Audioformat wiedergeben kann. Andernfalls ist der Wert auf `false` gesetzt.
`hasSMAF`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player Sounddaten im SMAF-Format (Synthetic music Mobile Application Format) wiedergeben kann. Andernfalls ist der Wert auf `false` gesetzt.
`hasDataLoading`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player dynamisch zusätzliche Daten über Aufrufe von `loadMovie()`, `loadMovieNum()`, `loadVariables()`, `loadVariablesNum()`, `XML.parseXML()`, `Sound.loadSound()`, `MovieClip.loadVariables()`, `MovieClip.loadMovie()`, `MovieClipLoader.loadClip()`, `LoadVars.load()` und `LoadVars.sendAndLoad()` laden kann. Andernfalls ist der Wert auf `false` gesetzt.
`has4WayKeyAS`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player den ActionScript-Code ausführen kann, der den mit den Tasten „Nach-rechts“, „Nach-links“, „Nach-oben“ oder „Nach-unten“ verknüpften `keyEvent`-Prozeduren zugeordnet ist. Andernfalls ist der Wert auf `false` gesetzt. Wenn eine der Vierwege-Tasten gedrückt wird und der Wert dieser Variablen `true` ist, sucht der Player zuerst nach einer Prozedur für die entsprechende Taste. Wenn keine Prozedur gefunden wird, erfolgt die Navigation über die Flash-Steuerelemente. Wird eine Ereignisprozedur gefunden, wird für diese Taste keine Navigationsaktion ausgeführt. Wenn beispielsweise eine Tastendruck-Prozedur für die Nach-unten-Taste gefunden wird, ist die Funktion zum Nach-unten-Navigieren deaktiviert.
`hasMouse`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player mausbezogene Ereignisse senden kann und auf `false`, wenn die Plattform keine Maus unterstützt.
`hasMappableSoftKeys`	Ermöglicht das Festlegen von Werten für Softtasten und das Auswerten der mit diesen Softtasten verbundenen Ereignisse.
`hasStylus`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn der Player auf den Eingabestift bezogene Ereignisse senden kann und auf `false`, wenn die Plattform keinen Eingabestift unterstützt. Das `onMouseMove`-Ereignis wird mit einem Eingabestift nicht unterstützt. Mithilfe dieses Flags kann vom SWF-Inhalt geprüft werden, ob dieses Ereignis unterstützt wird.
`hasCMIDI`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn die Plattform CMIDI-Sound unterstützt. Andernfalls ist der Wert auf `false` gesetzt.
`hasXMLSocket`	(neu in Flash Lite 2.1) Dieser Wert gibt an, ob die Hostanwendung XML-Sockets unterstützt.
`softKeyCount`	Gibt die Anzahl der auf der Plattform unterstützten Softtasten an.
`hasSharedObjects`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn durch den in dieser Anwendung wiedergegebenen Flash-Inhalt auf gemeinsamen Objekte von Flash Lite zugegriffen werden kann. Andernfalls ist der Wert auf `false` gesetzt.
`hasQWERTYKeyboard`	Ein schreibgeschützter boolescher Wert, der auf `true` gesetzt ist, wenn allen Tasten auf einer QWERTY-Standardtastatur (einschließlich der Rücktaste) ActionScript zugeordnet werden kann. Andernfalls ist der Wert auf `false` gesetzt.
`audioMIMETypes`	Eine schreibgeschützte Eigenschaft, die vom Sound-Objekt in ActionScript verwendet werden kann und ein Array mit MIME-Typen enthält, die von den Audiocodecs des Geräts unterstützt werden.

Eigenschaft	Beschreibung
`imageMIMETypes`	Eine schreibgeschützte Eigenschaft, die ein Array mit MIME-Typen enthält, die von den Bildcodecs des Geräts unterstützt werden und von der ActionScript-Funktion `loadMovie` verwendet werden können.
`videoMIMETypes`	Eine schreibgeschützte Eigenschaft, die ein Array mit MIME-Typen enthält, die von den Videocodecs des Geräts unterstützt werden und vom Video-Objekt in ActionScript verwendet werden können.
`MIMETypes`	Eine schreibgeschützte Eigenschaft, die ein Array aller MIME-Typen enthält, die vom Sound- und vom Video-Objekt unterstützt werden sowie von der ActionScript-Funktion `loadMovie()` verwendet werden können.

Sound-Klasse

ActionScript für Flash Lite 2.x und später unterstützt Geräte-Sound über die Sound-Klasse und über die Werte von System.capabilities. Die Sound-Klasse wird bei den in Flash Player 7 unterstützten nativen Sounds vollständig, bei Geräte-Sounds jedoch nur teilweise unterstützt.

Über die in Flash Lite 2.x hinzugefügte Unterstützung können Sie die Wiedergabe von Geräte-Sound mit der Darstellung von Animationen synchronisieren.

Hinweis: *Flash Lite 2.x und später unterstützt keine Aufnahmen von Sounddaten.*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der Sound-Klasse nur teilweise unterstützt werden.

Methode	Beschreibung	Unterstützung
`getPan()`	Gibt die im letzten `setPan()`-Aufruf festgelegte Balanceeinstellung als Ganzzahl zwischen -100 (links) und +100 (rechts) zurück. (Bei 0 wird der Sound gleichmäßig über den linken und rechten Kanal übertragen.) Mit dieser Einstellung wird die Links-Rechts-Balance des aktuellen und der zukünftigen Sounds in einer SWF-Datei gesteuert. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.	Teilweise unterstützt
`getTransform()`	Gibt die Soundtransformationsinformationen für das angegebene Sound-Objekt zurück, die beim letzten Aufruf von `Sound.setTransform()` eingestellt wurden. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.	Teilweise unterstützt
`loadSound()`	Lädt eine MP3-Datei in ein Sound-Objekt. Mit dem Parameter `isStreaming` können Sie angeben, ob der Sound ein Ereignissound oder ein Streaming-Sound ist. Ereignissounds müssen vollständig geladen sein, bevor sie wiedergegeben werden können. Sie werden in ActionScript von der Sound-Klasse verwaltet und reagieren auf alle Methoden und Eigenschaften dieser Klasse. **Einschränkungen:** Der Parameter „isStreaming“ wird in Flash Lite 2.x und später ignoriert.	Teilweise unterstützt

Methode	Beschreibung	Unterstützung
`setPan()`	Legt fest, wie der Sound über den linken und rechten Kanal (Lautsprecher) wiedergegeben wird. Legt bei Monosounds fest, auf welchem Lautsprecher (links oder rechts) der Sound wiedergegeben wird. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.	Teilweise unterstützt
`setTransform()`	Legt die Soundtransformations- bzw. Balanceinformationen für ein Sound-Objekt fest. Im Parameter `soundTransformObject` geben Sie ein Objekt an, das Sie mit der Konstruktormethode der generischen Object-Klasse erstellen. Dabei verwenden Sie Parameter, die die Verteilung des Sounds auf den rechten und linken Kanal (Lautsprecher) angeben. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.	Teilweise unterstützt
`setVolume()`	Stellt die Lautstärke für das Sound-Objekt ein. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.	Teilweise unterstützt

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften der Sound-Klasse aufgeführt, die in ActionScript für Flash Lite 2.x und später nur teilweise unterstützt werden.

Eigenschaft	Beschreibung
`duration`	Dauer eines Sounds in Millisekunden. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.
`position`	Anzahl der Millisekunden, die der Sound bereits wiedergegeben wurde. **Einschränkungen:** Wird für die Verwendung mit nativem Flash-Sound unterstützt, jedoch nicht für die Verwendung mit Geräte-Sound.

Zusätzliche Methoden in Flash Lite

In der folgenden Tabelle sind neue Methoden der Sound-Klasse aufgeführt, die nur in ActionScript für Flash Lite 2.x und später verfügbar sind.

Methode	Beschreibung
`getPan()`	Gibt den Wert des vorhergehenden Aufrufs von `setPan()` zurück. Diese Methode wird für Geräte-Sound nicht unterstützt.
`getTransform()`	Gibt den Wert des vorhergehenden Aufrufs von `setTransform()` zurück. Diese Methode wird für Geräte-Sound nicht unterstützt.
`loadSound()`	Lädt Sounddaten in allen Formaten in Flash Player. Diese Methode unterscheidet sich von der Flash Player 7-Implementierung, da die mit dieser Methode geladenen Sounddaten immer als Ereignissound behandelt werden. Der zweite Parameter der Methode wird daher immer ignoriert. Im folgenden Beispiel wird der Wert `true` ignoriert: `my_sound.loadSound("mysnd.mp3", true);`

Stage-Klasse

Bei der Stage-Klasse handelt es sich um eine Klasse der obersten Ebene, auf deren Methoden, Eigenschaften und Prozeduren Sie ohne Konstruktor zugreifen können. Verwenden Sie die Methoden und Eigenschaften dieser Klasse, um auf Informationen zu den Begrenzungen einer SWF-Datei zuzugreifen und diese zu bearbeiten.

Weitere Informationen finden Sie im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*.

Zusätzliche Eigenschaften in Flash Lite

In der folgenden Tabelle sind neue Eigenschaften der Stage-Klasse aufgeführt, die nur in ActionScript für Flash Lite 2.x und später verfügbar sind.

Eigenschaft	Beschreibung
`showMenu`	Blendet die Standardelemente im Flash Player-Kontextmenü ein oder aus.
`getTransform()`	Gibt den Wert des vorhergehenden Aufrufs von `setTransform()` zurück. Diese Methode wird für Geräte-Sound nicht unterstützt.
`loadSound()`	Lädt Sounddaten in allen Formaten in Flash Player. Diese Methode unterscheidet sich von der Flash Player 7-Implementierung, da die mit dieser Methode geladenen Sounddaten immer als Ereignissound behandelt werden. Der zweite Parameter der Methode wird daher immer ignoriert. Im folgenden Aufruf wird beispielsweise der Wert `true` ignoriert: `my_sound.loadSound("mysnd.mp3", true);`

TextField-Klasse

Textfelder sind visuelle Bühnenelemente, mit denen Sie Benutzern Text anzeigen können. Ähnlich wie bei einem Eingabetextfeld oder Textbereich in HTML können Sie mit Flash Textfelder als bearbeitbar (bzw. schreibgeschützt) festlegen, HTML-Formatierungen zulassen und die Unterstützung für mehrzeiligen Text aktivieren.

Mit der TextField-Klasse können Sie Textfelder erstellen. Alle dynamischen Textfelder und Eingabetextfelder in einer SWF-Datei sind Instanzen der TextField-Klasse. Sie können einem Textfeld im Eigenschafteninspektor einen Instanznamen zuweisen und es mit den Methoden und Eigenschaften der TextField-Klasse per ActionScript bearbeiten. TextField-Instanznamen erscheinen im Film-Explorer und im Dialogfeld „Zielpfad einfügen“ des Bedienfelds „Aktionen“.

Wenn Sie ein Textfeld dynamisch erstellen möchten, verwenden Sie anstelle des Operators `NEW` die Methode `MovieClip.createTextField()`.

Mit den Methoden der TextField-Klasse können Sie Text in dynamischen Textfeldern oder Eingabetextfeldern, die Sie beim Authoring oder zur Laufzeit erstellen, festlegen, auswählen und ändern.

In Flash Lite 2.x und später werden alle Eigenschaften der TextField-Klasse unterstützt, Geräteschriftarten können jedoch nur in Textfeldern angezeigt werden. Geräteschriftarten sind besondere Schriften in Flash, die nicht in einer SWF-Datei eingebettet sind. Flash Lite verwendet eine auf dem Mobilgerät installierte Schriftart, die der Geräteschriftart möglichst ähnlich ist. Da Schriftkonturen nicht eingebettet werden, fällt die SWF-Datei entsprechend kleiner aus. Da die Geräteschriftkonturen nicht eingebettet werden, wird jedoch der mit diesen Schriftarten erstellte Text auf Geräten, die über keine der Geräteschriftart entsprechende Schriftart verfügen, in anderer Weise angezeigt. Flash enthält drei Geräteschriften: _sans (ähnlich Helvetica oder Arial), _serif (ähnlich Times Roman) und _typewriter (ähnlich Courier).

Weitere Informationen zur TextField-Klasse finden Sie in folgenden Quellen:

- Kapitel 12, „Mit Text und Strings arbeiten", im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen", im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften der TextField-Klasse aufgeführt, die in ActionScript für Flash Lite 2.x und später nicht unterstützt werden.

Eigenschaft	Beschreibung
`antialiasType`	Gibt den für diese TextField-Instanz verwendeten Anti-Aliasing-Typ an.
`menu`	Verknüpft ein ContextMenu-Objekt mit einem Textfeld.
`mouseWheelEnabled`	Ein boolescher Wert, der angibt, ob Flash Player bei mehrzeiligen Textfeldern automatisch einen Bildlauf durchführen soll, wenn der Benutzer mit dem Mauszeiger in ein Textfeld klickt und das Mausrad dreht.
`restrict`	Gibt die Zeichen an, die ein Benutzer in das Textfeld eingeben kann.
`sharpness`	Die Schärfe der Glyphenkanten in dieser TextField-Instanz.
`styleSheet`	Fügt ein Stylesheet an das Textfeld an.
`thickness`	Gibt die Stärke der Glyphenkanten in dieser TextField-Instanz an.

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden der TextField-Klasse aufgeführt, die in ActionScript für Flash Lite 2.x und später nicht unterstützt werden.

Methode	Beschreibung
`getFontList`	Gibt die Namen von Schriftarten auf dem System, unter dem Flash Player ausgeführt wird, als Array zurück.

TextFormat-Klasse

Die TextFormat-Klasse enthält Zeichenformatierungsinformationen. Verwenden Sie diese Klasse zum Erstellen spezifischer Textformatierungen für Textfelder. Textformatierungen können auf statische und auf dynamische Textfelder angewendet werden. Einige Eigenschaften der TextFormat-Klasse stehen für eingebettete und Geräteschriften nicht zur Verfügung.

Mithilfe der TextFormat-Klasse können Sie einem Textfeld oder bestimmten Zeichen innerhalb eines Textfelds Formatierungen zuweisen. Textformatierungsoptionen sind beispielsweise Ausrichtung, Einzüge, Fettdruck, Farbe, Schriftgröße, Randbreite, Kursivschrift und Zeichenabstand. Textformatierungen können auf statische und auf dynamische Textfelder angewendet werden. Einige Eigenschaften der TextFormat-Klasse stehen für eingebettete und Geräteschriften nicht zur Verfügung.

Bevor Sie jedoch die Methoden eines TextFormat-Objekts aufrufen können, müssen Sie dieses Objekt mithilfe des Konstruktors `TextFormat()` erstellen.

Hinweis: *In Flash Lite 2.x und später werden die in der TextFormat-Klasse verfügbaren Formatierungsfunktionen teilweise unterstützt. Formatierungsfunktionen sind bei Verwendung von Geräteschriftarten nicht verfügbar.*

Die TextFormat-Klasse wird in Flash Lite 2.x und später teilweise unterstützt. TextFormat.font, TextFormat.bold und TextFormat.tabstop werden beispielsweise nicht unterstützt, wenn Sie Geräteschriftarten verwenden.

Weitere Informationen zur TextFormat-Klasse finden Sie in folgenden Quellen:

- Kapitel 12, „Mit Text und Strings arbeiten", im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen", im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften aufgeführt, die beim Erstellen von Flash-Inhalten mit ActionScript 2.0 nur teilweise unterstützt werden.

Eigenschaft	Beschreibung
`bold`	Ein boolescher Wert, der angibt, ob der Text fett formatiert ist. **Einschränkungen**: Nur bei Formatierung auf Absatzebene. Fettdruck kann nicht auf einzelne Zeichen angewendet werden.
`bullet`	Ein boolescher Wert, der angibt, ob der Text zu einer Liste mit Aufzählungszeichen gehört. **Einschränkungen:** Nur zur Verwendung mit eingebetteten Schriftarten.
`color`	Gibt die Textfarbe an. **Einschränkungen:** Nur bei Formatierung auf Absatzebene. Die Farbformatierung kann nicht auf einzelne Zeichen angewendet werden.
`font`	Ein String mit dem Namen der Schriftart für Text in diesem Textformat. **Einschränkungen:** In Flash Lite kann diese Eigenschaft nur bei eingebetteten Schriftarten verwendet werden. Diese Eigenschaft wird für Arabisch, Hebräisch und Thai nicht unterstützt.
`italic`	Ein boolescher Wert, der angibt, ob Text in diesem Textformat kursiv formatiert wird. **Einschränkungen:** Nur bei Formatierung auf Absatzebene. Die Kursivformatierung kann nicht auf einzelne Zeichen angewendet werden.

Eigenschaft	Beschreibung
`size`	Die Punktgröße von Text in diesem Textformat. **Einschränkungen:** Nur bei Formatierung auf Absatzebene. Unterschiedliche Schriftgrößen können nicht auf einzelne Zeichen angewendet werden.
`tabStops`	Gibt benutzerdefinierte Tabstopps als Array aus nicht negativen Ganzzahlen an. **Einschränkungen:** Nur zur Verwendung mit eingebetteten Schriftarten.
`underline`	Ein boolescher Wert, der angibt, ob der Text mit diesem Textformat unterstrichen ist (`true`) oder nicht (`false`). **Einschränkungen:** Nur bei Formatierung auf Absatzebene.

Video-Klasse

In Flash Lite 2.x können Sie gerätespezifische Videoformate verwenden. Darüber hinaus werden die folgenden Arten der Videowiedergabe unterstützt:

- In SWF-Dateien eingebettete Videodaten
- Als separate Dateien auf dem Gerät vorhandene Videodaten
- Per Streaming über das Netzwerk übertragene Videodaten (in Echtzeit)

Flash Lite 2.x unterstützt Gerätevideo. Gerätevideo wird in der veröffentlichten SWF-Datei im nativen Videoformat des Geräts gespeichert. Zur Wiedergabe von Gerätevideo übergibt Flash Lite die Videodaten an das Gerät. Dort werden diese dekodiert und dann wiedergegeben.

Hinweis: *Flash Lite 2.x ActionScript unterstützt weder die `NetConnection`-Klasse noch die `NetStream`-Klasse.*

Flash Lite 3.0 wurde um die Unterstützung für Flash Video (FLV) mithilfe von Versionen der On2- und Sorenson-Codecs, die für Mobilgeräte optimiert sind, erweitert. FLV-Video wird direkt im Flash Lite Player wiedergegeben, nicht vom Gerät. Auf diese Weise ist es irrelevant, ob Ihre Zielgeräte ein bestimmtes Videoformat unterstützen. Mithilfe der ActionScript-Klassen NetConnection und NetStream, die bisher in Flash Lite nicht verfügbar waren, können Sie die Wiedergabe von FLV-Video von einem lokalen Laufwerk oder einer HTTP-Adresse aus steuern. Diese Klassen werden in Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch* beschrieben. Das Streaming von Videodaten über eine RTMP-Verbindung an einen Flash Media Server wird ebenfalls in Flash Lite 3.0 unterstützt (RTMPT- und RTMPS-Verbindungen hingegen nicht). Weder die Camera-Klasse noch das Aufzeichnen von Video werden für Flash Video in Flash Lite unterstützt.

Flash Lite 3.0 enthält ebenfalls eine neue Eigenschaft in der Video-Klasse, `attachVideo`, die einen Video-Stream angibt, der innerhalb des Video-Objekts auf der Bühne angezeigt werden soll. Zum Steuern der Wiedergabe von FLV-Video (nicht Gerätevideo) verwenden Sie eher die Methoden der NetStream-Klasse als die der Video-Klasse. Um beispielsweise Gerätevideo anzuhalten, verwenden Sie die Methode `Video.pause`; um aber FLV-Video anzuhalten, verwenden Sie `NetStream.pause`.

Weitere Informationen zur Video-Klasse finden Sie in folgenden Quellen:

- Kapitel 15, „Mit Bildern, Audio und Video arbeiten“, im *ActionScript 2.0 in Flash – Arbeitshandbuch*
- Kapitel 2, „ActionScript-Klassen“, im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch*

Übersicht über Methoden

In der folgenden Tabelle sind die Methoden aufgeführt, die in ActionScript für Flash Lite 2.x und später von der Video-Klasse nicht unterstützt werden.

Methode	Beschreibung	Unterstützung
`clear`	Löscht das aktuelle im Video-Objekt angezeigte Bild. Hiermit können Sie beispielsweise Standby-Informationen anzeigen, ohne das Video-Objekt ausblenden zu müssen.	Nicht unterstützt

Übersicht über Eigenschaften

In der folgenden Tabelle sind die Eigenschaften der Video-Klasse aufgeführt, die in ActionScript für Flash Lite 2.x und neuere Versionen nicht unterstützt werden.

Eigenschaft	Beschreibung
`deblocking`	Gibt den Typ des Deblocking-Filters an, der bei der Nachbearbeitung auf dekodierte Videos angewendet wird. Es stehen zwei Deblocking-Filter zur Verfügung: einer im Sorenson-Codec und ein weiterer im On2 VP6-Codec.
`height`	Eine Ganzzahl, die die Höhe des Video-Streams in Pixel angibt.
`smoothing`	Gibt an, ob das Video beim Skalieren geglättet (interpoliert) werden soll. Die Funktion zum Glätten kann nur verwendet werden, wenn sich der Player im Modus für hohe Wiedergabequalität befindet.
`width`	Eine Ganzzahl, die die Breite des Video-Streams in Pixel angibt.

Zusätzliche Methoden in Flash Lite

Die Video-Klasse für Flash Lite 2.x und neuere Versionen verfügt über die folgenden neuen Methoden.

Methode	Beschreibung
`play()`	Öffnet eine Videoquelle und startet die Wiedergabe des Videos.
`close()`	Beendet die Wiedergabe des Videos, gibt den diesem Video-Objekt zugeordneten Speicher frei und löscht den Videobereich auf dem Bildschirm.
`stop()`	Beendet die Wiedergabe des Videos und stellt weiterhin das aktuelle Bild auf dem Bildschirm dar. Bei einem folgenden Aufruf von `Video.resume()` wird die Wiedergabe ab dem ersten Bild des Videos fortgesetzt.
`pause()`	Beendet die Wiedergabe des Videos und stellt weiterhin das aktuelle Bild auf dem Bildschirm dar. Bei einem folgenden Aufruf von `Video.resume()` wird die Wiedergabe ab der aktuellen Position fortgesetzt.
`resume()`	Setzt die Wiedergabe des Videos fort.

Die Video-Klasse für Flash Lite 3.0 fügt die folgende neue Methode hinzu.

Methode	Beschreibung
`attachVideo()`	Gibt einen Video-Stream (Quelle) an, der innerhalb des Video-Objekts auf der Bühne angezeigt werden soll.

Nicht oder teilweise unterstützte ActionScript-Elemente: details

In diesem Abschnitt werden die globalen Funktionen sowie die Eigenschaften, Konstanten, Operatoren, Anweisungen, Erweiterungen und Schlüsselwörter beschrieben, die von ActionScript für Macromedia Flash Lite 2.x-Software von Adobe oder von Adobe Flash Lite 3.x nur teilweise oder nicht unterstützt werden.

Mit fsCommand und fsCommand2 ausgegebene Befehle

Die globalen Funktionen `fsCommand()` und `fsCommand2()` ermöglichen die Kommunikation zwischen der SWF-Datei und dem Flash Player bzw. dem Programm, in dem der Flash Player ausgeführt wird (z. B. in einem Webbrowser).

In Flash Lite 2.x und neueren Versionen wurden die Macromedia Flash Player 7-Standardbefehle modifiziert und Befehle für eingebettete Geräte hinzugefügt. In Flash Lite 2.x und neueren Versionen wird auch die `fsCommand2()`-Funktion mit einem ähnlichen Funktionsumfang wie `fsCommand()` unterstützt. Dieser Befehl wird jedoch sofort ausgeführt und nicht bis zum Abschluss der Verarbeitung für das aufrufende Bild verzögert.

Weitere Informationen zur `fsCommand2()`-Funktion finden Sie im *Flash Lite 2.x und 3.x ActionScript-Referenzhandbuch.*

Nicht unterstützte Befehle

In der folgenden Tabelle sind die Befehle aufgeführt, die beim Erstellen von Flash Lite-Inhalten mit ActionScript2.0 von `fsCommand()` nicht unterstützt werden.

Befehl	Beschreibung
`quit`	Schließt den Projektor.
`fullscreen`	Bei Angabe von `true` wird Flash Player in den Vollbildmodus gesetzt. Mit `false` wird der Player auf die normale Menüansicht zurückgesetzt.
`allowscale`	Bei Angabe von `false` wird der Player so eingestellt, dass die SWF-Datei immer in ihrer ursprünglichen Größe dargestellt und nicht skaliert wird. Bei Angabe von `true` wird die SWF-Datei exakt in der Bildschirmgröße dargestellt.
`showmenu`	Bei Angabe von `true` werden sämtliche Kontextmenüelemente aktiviert. Bei Angabe von `false` werden alle Elemente des Kontextmenüs mit Ausnahme von „Über Flash Player“ und „Einstellungen“ ausgeblendet.
`exec`	Führt eine Anwendung von innerhalb des Projektors aus.
`trapallkeys`	Bei Angabe von `true` werden alle Tastenereignisse, einschließlich Tastaturbefehle, an die Prozedur `onClipEvent(keyDown/keyUp)` in Flash Player gesendet.

Befehlsübersicht

In der folgenden Tabelle sind die ActionScript-Befehle, -Funktionen und -Schlüsselwörter aufgeführt, die in Flash Lite 2.x und später entweder nicht oder nur teilweise unterstützt werden.

Befehl, Funktion oder Schlüsselwort	Beschreibung	Unterstützung
`asfunction`	Ein Protokoll für URLs in HTML-Textfeldern.	Nicht unterstützt
`fscommand()`	Eine Funktion, die die Kommunikation zwischen der SWF-Datei und Flash Player bzw. dem Programm, in dem Flash Player ausgeführt wird (z. B. in einem Webbrowser), ermöglicht.	Teilweise unterstützt
`on`	Präfix für Ereignisse, für die bei ihrem Auftreten bestimmte Prozeduren ausgeführt werden. **Einschränkung**: Es werden die Ereignisse `rollout`, `rollover` und `keyPress` unterstützt.	Teilweise unterstützt
`onClipEvent`	Ereignisprozedur. Löst Aktionen aus, die für eine bestimmte Instanz eines Movieclips definiert sind. **Einschränkungen**: Es werden die Ereignisse `press`, `load`, `unload`, `enterFrame`, `keyDown`, `keyup` und `data` unterstützt. Die Ereignisse `mouseDown`, `mouseUp` und `mouseMove` werden unterstützt, wenn `System.capabilities.hasMouse` oder `System.capabilities.hasStylus` den Wert `true` hat.	Teilweise unterstützt
`onUpdate`	Die `onUpdate`-Ereignisprozedur ist für die Direktvorschau einer Komponente definiert.	Nicht unterstützt
`print()`	Die `print()`-Funktion druckt einen Movieclip entsprechend den im Parameter angegebenen Begrenzungen.	Nicht unterstützt
`printAsBitmap()`	Eine Funktion, mit der ein Ziel-Movieclip als Bitmap gedruckt wird.	Nicht unterstützt
`printAsBitmapNum()`	Eine Funktion, mit der eine Stufe in Flash Player entsprechend den im Parameter angegebenen Begrenzungen (`bmovie`, `bmax` oder `bframe`) als Bitmap gedruckt wird.	Nicht unterstützt
`printNum()`	Eine Funktion, mit der eine Stufe in Flash Player entsprechend den im Parameter `boundingBox` angegebenen Begrenzungen (`bmovie`, `bmax` oder `bframe`) gedruckt wird.	Nicht unterstützt
`startDrag()`	Eine Funktion, durch deren Aufruf der Ziel-Movieclip während der Wiedergabe der SWF-Datei durch Ziehen verschoben werden kann. Es kann jeweils nur ein Movieclip gezogen werden. **Einschränkung**: Wird unterstützt, wenn die Maus- oder Eingabestiftschnittstelle unterstützt wird.	Teilweise unterstützt
`stopDrag()`	Eine Funktion, mit der der aktuelle Ziehvorgang beendet wird. **Einschränkung**: Wird unterstützt, wenn die Maus- oder Eingabestiftschnittstelle unterstützt wird.	Teilweise unterstützt
`updateAfterEvent()`	Eine Funktion, mit der die Anzeige (unabhängig von der für die SWF-Datei eingestellten Bildrate) aktualisiert wird, wenn sie in einer `onClipEvent()`-Prozedur oder als Teil einer an `setInterval()` übergebenen Funktion oder Methode aufgerufen wird.	Nicht unterstützt

Globale Eigenschaften

In der folgenden Tabelle sind die globalen ActionScript-Eigenschaften aufgeführt, die von Flash Lite 2.x und später nicht oder nur teilweise unterstützt werden.

Eigenschaften	**Beschreibung**	**Unterstützung**
`_droptarget`	Schreibgeschützte Eigenschaft, die in Schrägstrichnotation (/) den absoluten Pfad der Movieclip-Instanz zurückgibt, in der der `draggableInstanceName` (der Name der Movieclip-Instanz, die das Ziel eines Aufrufs der `startDrag`()-Funktion war) abgelegt wurde. Diese Eigenschaft gibt immer einen Pfad zurück, der mit einem Schrägstrich (/) beginnt.	Nicht unterstützt
`_highquality`	Globale Eigenschaft, die den Grad des in der aktuellen SWF-Datei verwendeten Anti-Aliasings angibt. Mit dieser Eigenschaft kann auch das Glätten von Bitmaps gesteuert werden. **Einschränkung**: Bitmapglättung wird in Flash Lite nicht unterstützt.	Teilweise unterstützt
`_url`	Schreibgeschützte Eigenschaft mit der URL der SWF-Datei, von der ein Movieclip heruntergeladen wurde.	Nicht unterstützt

Kapitel 11: Warnungen und Fehlermeldungen

In diesem Anhang sind sämtliche Informationen und Warnungen aufgeführt, die möglicherweise vom Gerät und vom Emulator in Adobe Device Central beim Testen einer Adobe Flash Lite-Anwendung ausgegeben werden.

Gerätefehlercodes

Während der Adobe Device Central-Emulator den Inhalt testet, können bei Bedarf Warnhinweise erzeugt werden. Bestimmte Warnhinweise werden nur im Emulator angezeigt und liefern Angaben zu tatsächlichen oder möglichen Fehlern. Warnhinweise des anderen Typs können sowohl im Emulator als auch auf dem tatsächlichen Gerät auftreten.

Warnhinweise des ersten Typs liefern nützliche Debugging-Informationen zur SWF-Datei. Wenn Ihre SWF-Datei z. B. Adobe ActionScript enthält, das von Flash Lite nicht unterstützt wird (bzw. von der Flash Lite-Version, die auf dem zurzeit ausgewählten Testgerät verfügbar ist), gibt der Emulator eine entsprechende Warnmeldung aus.

Der zweite Meldungstyp kann sowohl im Emulator als auch auf einem Gerät auftreten. Fehlermeldungen dieses Typs werden in einem Fehlerdialogfeld angezeigt, das der Benutzer schließen muss, um die Arbeit mit der Anwendung fortzusetzen. In der folgenden Abbildung sehen Sie ein Beispiel für ein Fehlerdialogfeld im Emulator:



Auf einem Gerät enthält das Fehlerdialogfeld den String „Problem mit Inhalt“ sowie eine Fehlernummer. Im Emulator enthält das Fehlerdialogfeld zusätzlich einen kurzen Fehlerstring. Außerdem gibt der Emulator eine ausführlichere Beschreibung des Fehlers im Bedienfeld „Ausgabe“ aus.

In der folgenden Tabelle sind alle Fehler aufgeführt, die im Flash Lite Player auftreten können. Sie finden hier jeweils die Fehlernummer, die im Fehlerdialogfeld angezeigte kurze Beschreibung sowie die im Bedienfeld „Ausgabe“ angezeigte ausführlichere Erläuterung.

Fehlernummer	Fehlerstring	Beschreibung und mögliche Ursachen
1	Zu wenig Speicher	Der Heap-Speicher des Emulators ist unzureichend. Der Emulator weist einer SWF-Datei 1 MB Arbeitsspeicher zu, sofern es keine anderen Vorgaben gibt.
2	Stapellimit erreicht	Der Emulator hat festgestellt, dass das Stapellimit erreicht oder überschritten wurde. Die Ursachen hierfür können vielfältig sein, z. B. mehrfach verschachtelte Movieclips oder komplizierte Vektorzeichnungen.
3	Beschädigte SWF-Datei.	Der Emulator hat festgestellt, dass die Daten der SWF-Datei beschädigt sind.
4	ActionScript hängt	Der Emulator hat festgestellt, dass die Ausführung von bestimmtem ActionScript-Code in der SWF-Datei zu lange dauert. Er hat daher die Ausführung des ActionScript-Codes gestoppt.
5	–	–
6	Fehlerhafte Bilddaten	Die SWF-Datei enthält ein Bild, das von Flash Lite oder dem nativen Bilddekodierer der Plattform nicht dekodiert werden konnte.
7	Fehlerhafte Sounddaten	In der SWF-Datei wurde versucht, einen Sound in einem nicht unterstützten Format zu laden, oder die Sounddaten weisen Fehler auf.
8	Stammfilm entladen	Dieser Fehler tritt auf, wenn die SWF-Stammdatei (Stufe 0) durch eine andere SWF-Datei ersetzt wird.
9	–	–
10	getURL-String zu lang	Die URL-String im Aufruf von getURL() ist zu lang.
11	Nicht genügend Speicher zum Dekomprimieren des Bilds verfügbar	Flash Lite verfügt nicht über ausreichenden Speicher zum Dekodieren eines Bilds in der SWF-Datei.
12	Fehlerhafte SVG-Daten	Flash Lite hat versucht, fehlerhafte SVG-Daten zu laden.
13	Nicht genügend Speicher beim Laden des Streams	Flash Lite verfügt nicht über ausreichenden Speicher zum Verarbeiten der Stream-Daten aus einer URL. Dieser Fehler kann zum Beispiel auftreten, wenn Sie versuchen, eine XML-Datei über das Netzwerk zu laden, die zu groß für die Verarbeitung durch Flash Lite ist. Versuchen Sie, die Datendatei in mehrere kleinere Dateien zu unterteilen und jede der Dateien einzeln zu laden, falls möglich.

Fehler- und Informationsmeldungen des Adobe Device Central-Emulators

Meldungen werden im Emulator in einem Ausgabe-Popupfenster und in Adobe Flash im Bedienfeld „Ausgabe“ angezeigt. In der folgenden Tabelle sind alle Meldungen und Informationen aufgeführt, die vom Adobe Device Central-Emulator ausgegeben werden.

Fehlercode	**Meldung**	**Beschreibung**
FTPA002	FSCommand wird ignoriert.	Der Emulator hat einen Funktionsaufruf des Typs `fscommand()` erkannt, der vom ausgewählten Testgerät nicht unterstützt wird. An der gerätespezifischen SWF-Datei werden keine Änderungen vorgenommen; dies ist lediglich eine Warnung.
FTPA003	loadVariables wird ignoriert.	Der Emulator hat einen Funktionsaufruf des Typs `loadVariables()` erkannt, der vom ausgewählten Testgerät und Inhaltstyp nicht unterstützt wird. An der gerätespezifischen SWF-Datei werden keine Änderungen vorgenommen; dies ist lediglich eine Warnung.
FTPA004	loadMovie wird ignoriert.	Der Emulator hat einen Funktionsaufruf des Typs `loadMovie()` erkannt, der vom ausgewählten Testgerät und Inhaltstyp nicht unterstützt wird. An der gerätespezifischen SWF-Datei werden keine Änderungen vorgenommen; dies ist lediglich eine Warnung.
FTPA005	Der Aufruf an GetURL für *URL* wurde ignoriert, weil es mehr als eine Anforderung pro Tastendruck gab.	In Flash Lite ist nur ein Funktionsaufruf des Typs `getURL()` pro Tastendruck zulässig. Der Emulator hat festgestellt, dass es mehrere Funktionsaufrufe des Typs `getURL()` gab. Nur der erste Aufruf wird verarbeitet; alle anderen werden ignoriert.
FTPA006	Der Aufruf an GetURL für *URL* wurde ignoriert, weil er mit keinem Tastendruck verbunden war.	Das gegenwärtig ausgewählte Testgerät und der Inhaltstyp verarbeiten nur `getURL()`-Funktionsaufrufe, die durch Drücken einer Taste auf dem Gerät ausgelöst werden. Der Emulator hat festgestellt, dass Ihre Anwendung einen Aufruf an `getURL()` übergeben hat, der nicht mit einem Tastendruck verbunden war.
FTPA007	getProperty oder setProperty nicht unterstützt bei: *movieclip-Eigenschaft*.	Flash Lite bietet keine Unterstützung für die angegebene Movieclip-Eigenschaft.
FTPA008	getProperty oder setProperty nicht vollständig unterstützt bei: *movieclip-Eigenschaft*.	Flash Lite bietet keine vollständige Unterstützung für die angegebene Movieclip-Eigenschaft. Weitere Informationen finden Sie im Eintrag für die entsprechende Eigenschaft im *Flash Lite 2.x and 3.0 ActionScript Language Reference*.
FTPA009	startDrag und stopDrag werden nicht unterstützt.	Der Emulator hat einen Funktionsaufruf des Typs `startDrag()` oder `stopDrag()` erkannt, der von Flash Lite nicht unterstützt wird.
FTPA014	getURL wird ignoriert.	Der Emulator hat einen Funktionsaufruf des Typs `getURL()` erkannt, der vom ausgewählten Testgerät und Inhaltstyp nicht unterstützt wird. An der gerätespezifischen SWF-Datei werden keine Änderungen vorgenommen; dies ist lediglich eine Warnung.

Fehlercode	**Meldung**	**Beschreibung**
FTPA015	Der Aufruf an loadMovie für *URL* wurde ignoriert, weil es mehr als eine Anforderung pro Tastendruck gab.	In Flash Lite ist nur ein Funktionsaufruf des Typs `loadMovie()` pro Tastendruck zulässig. Der Emulator hat festgestellt, dass es mehrere Funktionsaufrufe des Typs `loadMovie()` gab. Nur der erste Aufruf wird verarbeitet; alle anderen werden ignoriert.
FTPA016	Der Aufruf an loadMovie für *URL* wurde ignoriert, weil er mit keinem Tastendruck verbunden war.	Das gegenwärtig ausgewählte Testgerät und der Inhaltstyp verarbeiten nur `loadMovie()`-Funktionsaufrufe, die durch Drücken einer Taste auf dem Gerät ausgelöst werden. Der Emulator hat festgestellt, dass Ihre Anwendung einen Aufruf an `loadMovie()` übergeben hat, der nicht mit einem Tastendruck verbunden war.
FTPA017	Der Aufruf an loadVariables für *URL* wurde ignoriert, weil es mehr als eine Anforderung pro Tastendruck gab.	Ihre Anwendung hat während eines einzelnen Tastendruck-Ereignisses mehrere `loadVariables()`-Funktionsaufrufe übergeben. In Flash Lite ist nur ein `loadVariables()`-Aufruf pro Tastendruck zulässig. Nur der erste Aufruf wird verarbeitet; alle anderen werden ignoriert.
FTPA018	Der Aufruf an loadVariables für *URL* wurde ignoriert, weil er mit keinem Tastendruck verbunden war.	Das gegenwärtig ausgewählte Testgerät und der Inhaltstyp verarbeiten nur `loadVariables()`-Funktionsaufrufe, die durch Drücken einer Taste auf dem Gerät ausgelöst werden. Der Emulator hat festgestellt, dass Ihre Anwendung einen Aufruf an `loadVariables()` übergeben hat, der nicht mit einem Tastendruck verbunden war.
FTPA019	Der Aufruf an FSCommand mit den Argumenten *Befehl-Argumente* wurde ignoriert, weil es mehr als eine Anforderung pro Tastendruck gab.	In Flash Lite ist nur ein Funktionsaufruf des Typs `fscommand()` pro Tastendruck zulässig. Der Emulator hat festgestellt, dass es mehrere Funktionsaufrufe des Typs `fscommand()` gab. Nur der erste Aufruf wird verarbeitet; alle anderen werden ignoriert.
FTPA020	Der Aufruf an FSCommand mit den Argumenten *command-arguments* wurde ignoriert, weil er mit keinem Tastendruck verbunden war.	Das gegenwärtig ausgewählte Testgerät und der Inhaltstyp verarbeiten nur `fscommand()`-Funktionsaufrufe, die durch Drücken einer Taste auf dem Gerät ausgelöst werden. Der Emulator hat festgestellt, dass Ihre Anwendung einen Aufruf an `fscommand()` übergeben hat, der nicht mit einem Tastendruck verbunden war.
FTPE001	Die Taste wird nicht verarbeitet: *Tastenname* ASCII-Wert: *Wert*.	Der Emulator hat festgestellt, dass eine von Flash Lite nicht unterstützte Taste gedrückt wurde. Der Tastendruck wird ignoriert.
FTPE013	Eingabetextfelder werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Das aktuelle Testgerät und der Inhaltstyp bieten keine Unterstützung für Eingabetextfelder.
FTPS010	Streaming-Sound wird nicht unterstützt.	Das ausgewählte Testgerät und der ausgewählte Inhaltstyp bieten keine Unterstützung für Streaming-Sound.

Fehlercode	Meldung	Beschreibung
FTPS011	Es kann jeweils nur ein einzelner Sound abgespielt werden (Mischen nicht möglich).	Der Emulator hat festgestellt, dass die SWF-Datei mehrere gleichzeitig abgespielte Sounds enthält. Flash Lite bietet hierfür keine Unterstützung.
FTPS012	Ereignissound wurde ignoriert, weil er mit keinem Tastendruck verbunden war.	In Flash Lite 1.0 wird ein Sound nur wiedergegeben, wenn der Benutzer eine Taste auf dem Gerät drückt. (Diese Beschränkung trifft auf Flash Lite 2.0 und später nicht zu.)
FTPS021	Sound wird vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Das ausgewählte Testgerät und der ausgewählte Inhaltstyp bieten keine Unterstützung für Sound.
FTPS022	ADPCM-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat festgestellt, dass die SWF-Datei einen nativen (nicht vom Gerät stammenden) Sound enthält, der mit ADPCM komprimiert wurde. Dieser Sound wird vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt. An der gerätespezifischen SWF-Datei werden keine Änderungen vorgenommen; dies ist lediglich eine Warnung.
FTPS023	MP3-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat festgestellt, dass die SWF-Datei einen nativen (nicht vom Gerät stammenden) Sound enthält, der mit MP3 komprimiert wurde. Dieser Sound wird vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt. An der gerätespezifischen SWF-Datei werden keine Änderungen vorgenommen; dies ist lediglich eine Warnung.
FTPS024	MIDI-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MIDI-Gerätesound gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS025	PCM-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen nativen Flash-Sound gefunden, der mit PCM komprimiert wurde. Dieser Sound wird vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.
FTPS026	Debuggen wird von dem angegebenen Player zum Testen von Filmen nicht unterstützt.	Der Adobe Device Central-Emulator bietet keine Unterstützung für den Befehl Steuerung > Debuggen .
FTPS027	Sound-Bundle gefunden.	Der Emulator hat festgestellt, dass die SWF-Datei eine Sound-Bundle-Datei enthält.
FTPS028	Ungültiger FSCommand2 *Befehlsname*-Befehl gefunden.	Der angegebene `fscommand2()`-Befehl ist kein gültiger Befehlsstring. Eine Liste der gültigen `fscommand2()`-Befehle finden Sie unter der `fscommand2`-Funktion im *Flash Lite 2.x und 3.0 ActionScript-Referenzhandbuch*.
FTPS029	FSCommand2 *Befehlsname*-Befehl gefunden.	Der Emulator hat den angegebenen `fscommand2()`-Befehl erkannt.
FTPS030	FSCommand2 *Befehlsname*-Befehl wird vom Emulator nicht unterstützt; testen Sie ihn auf dem Gerät.	Der Emulator bietet keine Unterstützung für den angegebenen `fscommand2()`-Befehl. Sie müssen diese SWF-Datei auf einem Gerät testen, auf dem Flash Lite installiert ist, um festzustellen, ob der angegebene Befehl erwartungsgemäß funktioniert.

Fehlercode	Meldung	Beschreibung
FTPS031	Es wurden mehrere Instanzen von URL-Anforderungsaufrufen gefunden, aber nur eine Instanz pro Tastendruck/Bild ist zulässig.	In Flash Lite ist nur ein Funktionsaufruf des Typs `getURL()` pro Tastendruck oder Bild zulässig. Der Emulator hat festgestellt, dass es mehrere Funktionsaufrufe des Typs `getURL()` gab. Nur der erste Aufruf wird verarbeitet, alle anderen werden ignoriert.
FTPS032	Aufruf an GetURL(*URL*) gefunden; möglicherweise gelten Einschränkungen.	Der Emulator hat einen Funktionsaufruf des Typs `getURL()` gefunden, für den bei der Wiedergabe auf dem ausgewählten Gerät möglicherweise einige Runtime-Einschränkungen gelten. Testen Sie die SWF-Datei direkt auf einem Gerät, um festzustellen, ob der Befehl erwartungsgemäß funktioniert.
FTPS033	Aufruf an loadVariables(*URL*) gefunden; möglicherweise gelten Einschränkungen.	Der Emulator hat einen Funktionsaufruf des Typs `loadVariables()` gefunden, für den bei der Wiedergabe auf dem ausgewählten Gerät möglicherweise einige Runtime-Einschränkungen gelten. Testen Sie die SWF-Datei auf einem Gerät, um sicherzustellen, dass der Befehl erwartungsgemäß funktioniert.
FTPS034	Aufruf an FSCommand (*Befehlsname*) gefunden; möglicherweise gelten Einschränkungen.	Dies ist lediglich eine Warnung, dass möglicherweise nicht alle Geräte und Flash Lite-Inhaltstypen den Befehl `fscommand()` in der Anwendung unterstützen. Testen Sie die SWF-Datei auf einem Gerät, um sicherzustellen, dass der Befehl erwartungsgemäß funktioniert.
FTPS035	Aufruf an loadMovie(*URL*) gefunden; möglicherweise gelten Einschränkungen.	Der Emulator hat einen Funktionsaufruf des Typs `loadMovie()` gefunden, für den bei der Wiedergabe auf dem ausgewählten Gerät möglicherweise einige Runtime-Einschränkungen gelten. Testen Sie die SWF-Datei auf einem Gerät, um sicherzustellen, dass der Befehl erwartungsgemäß funktioniert.
FTPS036	Im Sound-Bundle wurden *N* Kilobyte *Gerätesound-Sound* gefunden.	Der Emulator gibt für jeden in einem Sound-Bundle gefundenen Sound den Typ (z. B. MIDI oder SMAF) und die Größe an.
FTPS037	SMAF-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen SMAF-Gerätesound gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS038	Der Aufruf an StartVibrate wurde ignoriert, weil es mehr als eine Anforderung pro Tastendruck oder Ereignis gab.	In Flash Lite ist nur ein Funktionsaufruf des Typs `fscommand2("StartVibrate")` pro Tastendruck oder Bild zulässig. Der Emulator hat festgestellt, dass es mehrere Funktionsaufrufe dieses Typs gab. Nur der erste Aufruf wird verarbeitet; alle anderen werden ignoriert.
FTPS039	FSCommand2 SetInputTextType (*Befehl-Argumente*) gefunden, aber vom Emulator nicht unterstützt. Führen Sie den Test auf dem Gerät durch.	Der Befehl `SetInputTextType` wird vom Emulator nicht unterstützt. Sie müssen ihn direkt auf einem Gerät testen.

Fehlercode	Meldung	Beschreibung
FTPS048	Vierwege-Navigation wird bei diesem Gerät nicht unterstützt.	Das gegenwärtige Testgerät und der Inhaltstyp unterstützen die Zweiwege-Navigation. Sie haben auf der Fünfwege-Tastatur des Emulators die Taste <Nach-links > oder <Nach-rechts > gedrückt. Diese beiden Tasten werden bei der Zweiwege-Navigation nicht unterstützt. Weitere Informationen finden Sie unter „Standardnavigationsmodi“ auf Seite 38.
FTPS049	Vierwege-Navigation mit Rücksprungfunktion wird bei diesem Gerät nicht unterstützt.	Das gegenwärtige Testgerät und der Inhaltstyp unterstützen die Vierwege-Navigation. Sie haben eine der Pfeiltasten auf dem Gerät gedrückt, obwohl es in Richtung der jeweiligen Pfeiltaste keine Objekte auf der Bühne gibt, die den Fokus erhalten können. Weitere Informationen finden Sie unter „Standardnavigationsmodi“ auf Seite 38.
FTPS050	Generische MFI-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen generischen MFi-Gerätesound gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS051	Nicht unterstütztes Mausereignis (*Ereignisname*) gefunden.	Das angegebene Mausereignis wird vom ausgewählten Testgerät und Inhaltstyp nicht unterstützt.
FTPS067	SMAF(MA-2)-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen SMAF (MA2)-Gerätesound gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS068	SMAF(MA-3)-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen SMAF (MA-3)-Gerätesound gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS069	SMAF(MA-5)-Sounds werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen SMAF (MA-5)-Gerätesound gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS070	MFI-Sounds mit Fujitsu-Erweiterung werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MFi-Gerätesound mit einer Fujitsu-Erweiterung gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS071	MFI-Sounds mit Mitsubishi-Erweiterung werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MFi-Gerätesound mit einer Mitsubishi-Erweiterung gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS072	MFI-Sounds mit NEC-Erweiterung werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MFi-Gerätesound mit einer NEC-Erweiterung gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS073	MFI-Sounds mit Panasonic-Erweiterung werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MFi-Gerätesound mit einer Panasonic-Erweiterung gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.

Fehlercode	**Meldung**	**Beschreibung**
FTPS074	MFI-Sounds mit Sharp-Erweiterung werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MFi-Gerätesound mit einer Sharp-Erweiterung gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS075	MFI-Sounds mit Sony-Erweiterung werden vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt.	Der Emulator hat einen MFi-Gerätesound mit einer Sony-Erweiterung gefunden, der vom ausgewählten Inhaltstyp auf diesem Gerät nicht unterstützt wird.
FTPS099	Druckbefehle werden nicht unterstützt.	Ihre Anwendung enthält einen der ActionScript-Drucken-Befehle (z. B. `print()` oder `printAsBitmap()`), die nicht von Flash Lite unterstützt werden.
FTPS100	*Gerätesound-Sound* wurde im Sound-Bundle gewählt.	Gibt den Namen des Gerätesounds in einem Sound-Bundle an, der vom Emulator wiedergegeben wurde.
FTPS101	Keines der im Sound-Bundle enthaltenen Formate wird von diesem Gerät unterstützt.	Gibt an, dass keiner der im Sound-Bundle enthaltenen Gerätesounds vom ausgewählten Inhaltstyp auf diesem Gerät unterstützt wird.
FTPS102	Fehler bei SMAF-Soundwiedergabe.	Der Emulator konnte den SMAF-Sound nicht abspielen.
FTPS105	Diese SWF-Datei liegt nicht im Flash Lite-Format vor.	Ihre Anwendung hat versucht, eine SWF-Datei zu laden, deren Version nicht dem Flash Lite-Format entspricht. Flash Lite kann nur SWF-Dateien laden, die ebenfalls aus Flash Lite stammen oder das Flash 4-Format aufweisen.
FTPS106	Mausereignis (*Ereignisname*) wurde ignoriert, weil es nicht von einem Tastendruck ausgelöst wurde.	Der Emulator hat ein Mausereignis über einer Taste in der Flash Lite-Anwendung erkannt. Das aktuelle Testgerät bietet keine Unterstützung für einen Stift oder eine Touchscreen-Benutzeroberfläche. Sie können die Schaltflächen auf dem Bildschirm nur über die Emulator-Tastatur bzw. über entsprechende Tastaturbefehle bedienen.
FTPS107	Die Taste wird nicht verarbeitet: *Gerätetaste*. Aktivieren Sie diese Taste mit FSCommand2 SetSoftKeys.	Sie haben eine der Softtasten auf der Emulator-Tastatur gedrückt, ohne zuerst den Befehl `SetSoftKeys` aufzurufen. Weitere Informationen finden Sie unter „Softtasten verwenden“ auf Seite 51.
FTPS108	Ungültigen FSCommand (*Befehlsname*) gefunden.	Der angegebene `fscommand()`-Befehl ist kein gültiger Befehlsstring.
FTPS109	FSCommand (*Befehlsname*) wird vom Emulator nicht unterstützt und sollte daher auf dem Gerät getestet werden.	Der Emulator bietet keine Unterstützung für den angegebenen `fscommand()`-Befehl. Testen Sie diese SWF-Datei auf einem Gerät, auf dem Flash Lite installiert ist, um festzustellen, ob der angegebene Befehl erwartungsgemäß funktioniert.
FTPS110	Softtasten werden im Flash Lite 1.0 Player nicht unterstützt.	Der Emulator hat festgestellt, dass Sie eine seiner Softtasten gedrückt haben. Allerdings wurde in den Veröffentlichungseinstellungen der SWF-Datei die Version Flash Lite 1.0 ausgewählt. Flash Lite 1.0 bietet keine Unterstützung für Softtasten.

Stichwortverzeichnis